MONASH University

Anomaly Detection in Streaming Time Series Data

Priyanga Dilini Talagala

B.Sc. (Hons), University of Sri Jayewardenepura, Sri Lanka

A thesis submitted for the degree of Doctor of Philosophy at

Monash University in 2019

Department of Econometrics and Business Statistics

Contents

Copyright Notice

Abstract

Anomaly detection has wide variations in problem formulations, which demand different analytical approaches. Despite the ever-increasing attention and resources devoted to the area of anomaly detection, some challenges are not supported by the existing frameworks and algorithms. This thesis reduces this gap by introducing three new algorithms for anomaly detection with special reference to their capabilities, competitive features and target applications.

This thesis offers four fundamental contributions. First, it proposes an improved algorithm for anomaly detection in high-dimensional data. It outperforms the state-of-the-art methods in many examples in terms of both accuracy and computational efficiency, while retaining a valid probabilistic interpretation for the anomalous threshold. Further, many existing algorithms have been specifically developed for the batch scenario, where it is assumed that all available data have been collected prior to analysis. However, with the recent rapid advances in data collection technology, streaming data are now becoming increasingly important and pose various challenges due to nonstationarity, noisy signals, large volume, high velocity, incomplete events and online support. To meet these challenges, as a second contribution, the thesis proposes another algorithm that provides early detection of anomalies within a large collection of streaming time series data. This algorithm includes a novel approach that adapts to nonstationarity. Third, it proposes a new algorithm to detect anomalies, caused by technical issues, in water-quality data from in situ sensors. Fourth, with the aim of facilitating reproducible research, the first, second and third algorithms are implemented in three open source R packages: `stray`, `oddstream` and `oddwater`, respectively. Using various synthetic and real datasets, this thesis demonstrates the wide applicability and usefulness of the three algorithms.

In `stray`, an anomaly is defined as an observation that deviates markedly from the majority with a large distance gap. This improved unsupervised algorithm for high-dimensional data is based on distance measures and the extreme value theory. In `oddstream`, an anomaly is defined as an observation that is very unlikely, given the recent distribution of a given system. In this algorithm, a boundary for the system's typical behaviour is calculated using the extreme value theory. Then, a sliding window is used to test newly arrived data. The model uses time series features as inputs and a density-based comparison to locate nonstationarity. `Oddwater` involves an application where anomaly detection is performed using turbidity, conductivity and river level data collected from rivers flowing into the Great Barrier Reef lagoon, Australia.

The algorithm, `stray`, which is specially designed for high-dimensional data, addresses the limitations of the state-of-art-method, the `HDoutliers` algorithm. Using various applications, this thesis demonstrates how `stray` can be used to detect anomalies in other data types, such as temporal data and streaming data. Applications of `oddstream` with data obtained using fibre optic cables showed that the framework has the ability to provide early detection of anomalies in large streaming nonstationary data. `Oddwater` successfully identified abrupt changes caused by technical outliers in water-quality sensors, while maintaining very low false detection rates.

Publications During Enrolment

This thesis by publication is built around four articles which are at different stages of publication.

1. Chapter 2 has been submitted to *Computational Statistics & Data Analysis* for possible publication.

   Talagala, P. D., Hyndman, R. J. & Smith-Miles, K. (2019). Anomaly Detection for High Dimensional Data. *arXiv preprint arXiv:1908.04000.*

2. Chapter 3 has been accepted for publication in the *Journal of Computational and Graphical Statistics* and is currently in press.

   Talagala, P. D., Hyndman, R.J., Smith-Miles, K., Kandanaarachchi, K., & Muñoz, M.A., (2019) Anomaly Detection in Streaming Nonstationary Temporal Data, *Journal of Computational and Graphical Statistics*, DOI: 10.1080/10618600.2019.1617160.

   I won the ACEMS Business Analytics prize 2018 for this work.

3. Chapter 4 has been revised and resubmitted to the *Water Resources Research* for possible publication.

   Talagala, P. D., Hyndman, R. J., Leigh, C., Mengersen, K., & Smith-Miles, K. (2019). A feature-based framework for detecting technical outliers in water-quality data from in situ sensors. *arXiv preprint arXiv:1902.06351.*

4. Chapter 5 is published in the *Science of the Total Environment.*

   Leigh, C., Alsibai, O., Hyndman, R. J., Kandanaarachchi, S., King, O. C., McGree, J. M., Neelamraju, C., Strauss, J., Talagala, P.D., Turner, R.D., Mengersen, K.,

& Peterson, E.E. (2019). A framework forautomated anomaly detection in high frequency water-quality data from in situ sensors. *Science of the Total Environment* 664, 885-898.

The contribution in Chapters 2, 3 and 4 of this thesis were presented at the following events:

- 37th International Symposium on Forecasting 2017, Cairns, Australia.
- Young Statisticians Conference 2017, Tweed Heads NSW, Australia.

  A Statistical Society of Australia travel award grant was received to attend the conference.
- Young Stats Showcase hosted by the Statistical Society of Australia, Victorian Branch, Australia, in September 2017.
- 38th International Symposium on Forecasting 2018, Boulder, Colorado, USA.

  An International Institute of Forecasters travel award grant was received to attend the conference.
- 2018 Joint Statistical Meetings (JSM2018), Vancouver, British Columbia, Canada
- useR! 2018, Brisbane, Australia.
- Joint International Society for Clinical Biostatistics and Australian Statistical Conference 2018 (ISCB ASC18), Melbourne, Australia.

  I won the EJP Pitman Young Statisticians Prize 2018, Merit Award 'for the outstanding talk presented by a *Young Statistician* at an Australian Statistical Conference'.
- 39th International Symposium on Forecasting, Thessaloniki, Greece.

  An International Institute of Forecasters travel award grant was received to attend the conference.
- useR! 2019, Toulouse, France.

  A conference diversity scholarship was received to attend the conference.

Declaration

I hereby declare that this thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

This thesis includes 2 original papers published in peer reviewed journals and 2 submitted publications. The core theme of the thesis is anomaly detection in streaming time series data. The ideas, development and writing up of all the papers (with the exception of Chapter 5) in the thesis were the principal responsibility of myself, the student, working within the Department of Econometrics and Business Statistics, Monash University, under the supervision of Professor Rob J. Hyndman and Professor Kate Smith-Miles.

The inclusion of co-authors reflects the fact that the work came from active collaboration between researchers and acknowledges input into team-based research.

In the case of Chapter 2-5 my contribution to the work involved the following:

Thesis Chapter	Publication Title	Status (published, in press, accepted or returned for revision)	Nature and % of student contribution	Co-author name(s) Nature and % of Co-author's contribution	Co-author(s) Monash student Y/N
2	Anomaly Detection for High Dimensional Data	Submitted	Formulating the approach, construction of research design, implementation, data analysis and interpretation, software development, writing the first draft (80%)	1) Rob J. Hyndman, input into manuscript (10%) 2) Kate Smith-Miles, input into manuscript (10%)	N
3	Anomaly Detection in Streaming Non-stationary Temporal Data	Published	Formulating the approach, construction of research design, implementation, data analysis and interpretation, software development, writing the first draft (80%)	1) Rob J. Hyndman, input into manuscript (10%) 2) Kate Smith-Miles, input into manuscript (5%) 3) Sevvandi Kandanaarachchi and Mario A. Muñoz, input into manuscript (5%)	N
4	A feature-based procedure for detecting technical outliers in water-quality data from in situ sensors	Revised and resubmitted	Formulating the approach, construction of research design, implementation, data analysis and interpretation, software development, writing the first draft (80%)	1) Rob J. Hyndman, input into manuscript (10%) 2) Catherine Leigh, Kerrie Mengersen and Kate Smith-Miles, input into manuscript (10%)	N
5	A framework for automated anomaly detection in high frequency water-quality data from in situ sensors	Published	Formulating the feature based anomaly detection approach: implementation, data analysis and interpretation, software development, writing the first draft of the feature based approach related parts. Developing a Shiny web application to explore data. Data preprocessing. (30%)	(1) Catherine Leigh, input into manuscript (40%) 2) Rob J. Hyndman, input into manuscript (10%) 3) Other co-authors, input into manuscript (20%)	N

I have not renumbered sections of submitted or published papers in order to generate a consistent presentation within the thesis.

Student name: Priyanga Dilini Talagala **Date:**19.08.2019

Acknowledgements

Over the past few years, with my experience on anomaly detection, I realised how important the support from the surrounding points is for a point to stand out as an anomaly, just like my support system around me supported me to make this thesis a reality. Now, it is my great pleasure to thank everyone who made this thesis possible.

I am deeply grateful to my supervisor Professor Rob J. Hyndman. Your valuable guidance and your way of thinking about, and doing, research continue to inspire me and shaped my thinking about all facets of this research. My research style, my research focus, my research toolkit – they all have their own roots in your mentorship. I am also very grateful to my co-supervisor Professor Kate Smith-Miles from University of Melbourne, Australia, who always raised important questions I never would have considered. I am truly blessed to have you two throughout this very special journey. Your influence on my thinking about research from different angles and boosting my confidence is beyond estimation.

Chapters 3 and 4 are based on the collaborative research project carried out with the Queensland University of Technology and the Queensland Department of Environment and Science, Great Barrier Reef Catchment Loads Monitoring Program. I consider myself very lucky to have had such a great opportunity to work on this project with many wonderful mentors, including Professor Kerrie Mengersen, Dr Erin Peterson and Dr. Catherine Leigh. Thank you for your thoughtful critiques, support, guidance, encouragement and generous hospitality during my visit to Queensland University of Technology, Australia, in April 2018. It was pure joy working with you all.

I was able to learn so much at, Monash University because of financial support from the Monash Graduate Scholarship and the Faculty Graduate Research scholarship. This

assistance has provided me many unique opportunities that I will always be thankful for. I am also thankful to ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS) for funding my visit to Queensland to work on the water-quality project and the Australian Research Council through the Linkage Project LP160101885 for funding the research study in Chapter 1 of this thesis. I am also thankful to the Queensland Department of Environment and Science; in particular, the Great Barrier Reef Catchment Loads Monitoring Program for the data, and the staff from Water Quality and Investigations for their input on Chapters 3 and 4 of this thesis. Further, this research was supported in part by the Monash eResearch Centre and eSolutions Research Support Services through the use of the MonARCH (Monash Advanced Research Computing Hybrid) HPC Cluster. I want to thank them for their valuable contribution.

I am also very thankful to my thesis committee, Professor Gael Martin, Professor Farshid Vahid, Professor George Athanasopoulos and Professor Xueyan Zhao, for their valuable feedback and suggestions. I am also grateful to David Hill and the many anonymous reviewers who read the manuscripts presented in Chapters 3, 4 and 5 and generously provided insights and suggestions to improve our work. I am also thankful to all my co-authors for their valuable contribution and collaborations. Special thanks to Cathy Morgan and Elite Editing for helping in copy editing and clarifying the manuscripts. The editorial intervention by Elite Editing was restricted to Standards D and E of the Australian Standards for Editing Practice.

I take this opportunity to thank the Statistical Society of Australia (SSA) for the travel grant to present at the Young Statisticians Conference 2017, Tweed Heads, NSW, Australia; the International Institute of Forecasters for the travel grant to present at the 38th International Symposium on Forecasting, Boulder, Colorado, USA and the travel grant to present at the 39th International Symposium on Forecasting, Thessaloniki, Greece; useR! 2019 for the travel grant to present at useR! 2019, Toulouse, France; and Monash EBS travel grant to present at JSM 2018, Vancouver, Canada. Each of these conferences is a unique experience and allowed me to share my research findings with a wider community. These conferences helped me significantly to attract the attention of peers and experts in

my field, and the questions, comments and suggestions I received from both academic and industry-oriented researchers helped me considerably in improving my work.

During my stay at Monash, the department administrative staff members, especially Clare Livesey, were very supportive and positive regarding every request. The Monash graduate research team was also very helpful and timely with all requests and I want to thank them for their assistance.

Special thanks to my sister, Thiyanga Talagala, who was there with, and for me, every step of the way, sharing many hats as siblings, schoolmates, college mates, workmates and office mates. I was extremely lucky to have an opportunity to have her by my side and share our latest and most precious hats as academic sisters and batch mates together at Monash University. Deeply grateful to our parents whom I love, admire, respect and find comfort in beyond words; you are a pillar of strength for me. All three of you have had a bigger influence on this thesis than you might realise.

Chapter 1

Introduction

Anomaly detection is an important research topic that has been explored within diverse research areas and application domains. The presence of anomalies in data can lead to biased parameter estimation, model misspecification and misleading results if classical analysis techniques are blindly applied (Abuzaid, Hussin, and Mohamed, 2013; Ben-Gal, 2005). Conversely, anomalies themselves can be the main carriers of significant and often critical information and the identification of these critical points can be the main purpose of many investigations in fields such as fraud detection (e.g., credit card frauds and network intrusion), object tracking (e.g., flight tracking), system health monitoring (e.g., machine breakdown and power cable leakages) and environmental monitoring (e.g., water quality, bushfire, earthquake and volcanic eruption) (Gupta et al., 2014). Further, owing to rapid advances in data collection technology it has become increasingly common for organisations to be dealing with data that stream in large quantities. Therefore, the overall focus of this thesis is on detecting anomalies in streaming time series data.

1.1 Background

This section reviews the background work on anomaly detection for streaming time series data and lays the foundation for the work presented in Chapters 2–5

1.1.1 Definitions Found in the Literature

Solutions to the problem of detecting unusual behaviours in systems of interest can be influenced heavily by the way in which anomalies are defined. Three terms are used commonly and interchangeably in the literature to describe work related to the topic: *novelty* (Clifton, Hugueny, and Tarassenko, 2011; Hugueny, 2013), *anomaly* (Hyndman, Wang, and Laptev, 2015; Kumar et al., 2016) and *outlier* (Schwarz, 2008; Wilkinson, 2018). However, Faria et al. (2016) differentiate between these three terms, using the terms anomaly and outlier to refer to the idea of an undesired pattern but novelty to refer to the emergence of a new concept that needs to be incorporated into the typical behaviour of the system. In line with this view, Chandola, Banerjee, and Kumar (2009) define an anomaly as a pattern in the data that does not conform to the expected behaviour but a novelty as an unobserved pattern that is typically incorporated into the model of the typical behaviour of a given system when it is detected. However, Gama (2010) points out that a substantial number of examples is required as evidence of the appearance of a novelty before it should be incorporated into the model of the typical behaviour of a given system. Thus, the sparse examples that differ considerably from the 'typical' behaviour can all be considered anomalies or outliers, since there is no guarantee that they represent a new 'typical' behaviour of the system (Faria et al., 2016). Lavin and Ahmad (2015) define anomalies in streaming data, with respect to their past behaviour, as patterns that do not conform to the past behaviours of the system. As a result, a new behaviour may be anomalous at first, but it ceases to be anomalous if the new 'typical' pattern continues to exist, and ultimately ends up being a novelty rather than an anomaly or an outlier.

Grubbs (1969) defines an anomaly as an observation that deviates markedly from other members of the sample. However, this deviation can be defined in terms of either distance or density. Burridge and Taylor (2006), Wilkinson (2018) and Schwarz (2008) have all proposed methods for anomaly detection by defining an anomaly in terms of distance. In contrast, Hyndman (1996), Clifton, Hugueny, and Tarassenko (2011) and Hugueny (2013) have proposed methods that define an anomaly with respect to either the density or the chance of the occurrence of observations.

1.1.2 Representation of Time Series

According to Fulcher and Jones (2014), the representation of time series is twofold: instance-based and feature-based.

The instance-based representation of time series is the most straightforward and has been used by many researchers in the data mining community. Under this representation, if two time series are to be compared, a distance metric between the two time series is defined that leads to a direct comparison of the ordered values of the two time series. The methods proposed by Wilkinson (2018), Clifton, Hugueny, and Tarassenko (2011) and Hugueny (2013) are all based on this representation of time series.

In contrast to the instance-based representation of time series, the feature-based representation of time series involves representing a given time series in terms of its properties, measured using different statistical operations, thereby transforming a temporal problem into a static problem (Fulcher, Little, and Jones, 2013). After extracting features, further analysis is based on these extracted features. Thus, this representation can allow an algorithm to compare time series of different lengths and/or starting points, because it can transform time series of any length or starting point into a vector of features of a fixed size. Recently, researchers such as Wang, Smith, and Hyndman (2006), Fulcher (2012) and Hyndman, Wang, and Laptev (2015) have paid a considerable amount of attention to the feature-based representation of time series, since it helps to reduce the dimension of the original multivariate time series problem via features that encapsulate the dynamic properties of the individual time series efficiently.

1.1.3 Extreme Value Theory

The algorithms proposed in Chapters 2, 3 and 4 are based on th extreme value theory, a branch of probability theory that relates to the behaviour of extreme order statistics in a given sample (Galambos, Lechner, and Simiu, 2013). In contrast to traditional data analysis, where the primary focus is on the observations in the central region of the distribution, extreme value theory focuses primarily on modelling the distribution of extreme order statistics in a given sample (Pinto and Garvey, 2016; Clifton, 2009). The

central limit theorem is one of the most striking limit theorems in statistics. Its ability to approximate the distribution of the sample mean irrespective of the parent distribution of the original random variable is the property that makes this theorem so remarkable (Coles, 2001). Analogous arguments are used in the extreme value theory to approximate distributions of extreme order statistics in a given sample.

1.1.4 Key Results of Classical Extreme Value Theory

Consider a set of m independent and identically distributed (iid) data, $\mathbf{X} = \{x_1, x_2, \ldots, x_m\}$, which has its own cumulative distribution function (CDF), F, and an associated probability density function (pdf), f. In classical extreme value theory, $x_i \in \Re$ (univariate). Let $X_{max} = max(\mathbf{X})$ and $X_{min} = min(\mathbf{X})$. The extreme value theory focuses on the statistical behaviour of these quantities. Hereafter, the discussion will focus on X_{max} (X_{min} will be referred to only when necessary), because it simplifies the discussion, but a similar argument can be applied to X_{min} as well.

The distribution of X_{max} can be investigated by taking several random samples of size m from a given distribution, recording the maximum of each sample and constructing a density plot of the maxima. Figure 1.1 (reproduced from Hugueny (2013), p. 87) shows the empirical distributions of minima and maxima for the standard Gaussian distribution (left), and of maxima for the standard exponential distribution (right) for series of sizes m. Each density plot is based on 10^6 data points. Consider the case of $m = 1$, where we observe only one data point from f in each trial. The corresponding density plot approximates the generative distribution f, because the maximum of a singleton set $\{x\}$ is simply x. However, the density plots for maxima move to the right as m increases, implying that the expected location of the sample maximum on the x-axis increases as more data are observed from f. Let H^+ denote the distribution function of X_{max}. This is termed the *extreme value distribution* (EVD), because it describes the expected location of the maximum of a sample of size m generated from f (Clifton, Hugueny, and Tarassenko, 2011). The Fisher–Tippett Theorem (Fisher and Tippett, 1928), which is the basis of classical extreme value theory, explains the possibilities for this H^+.

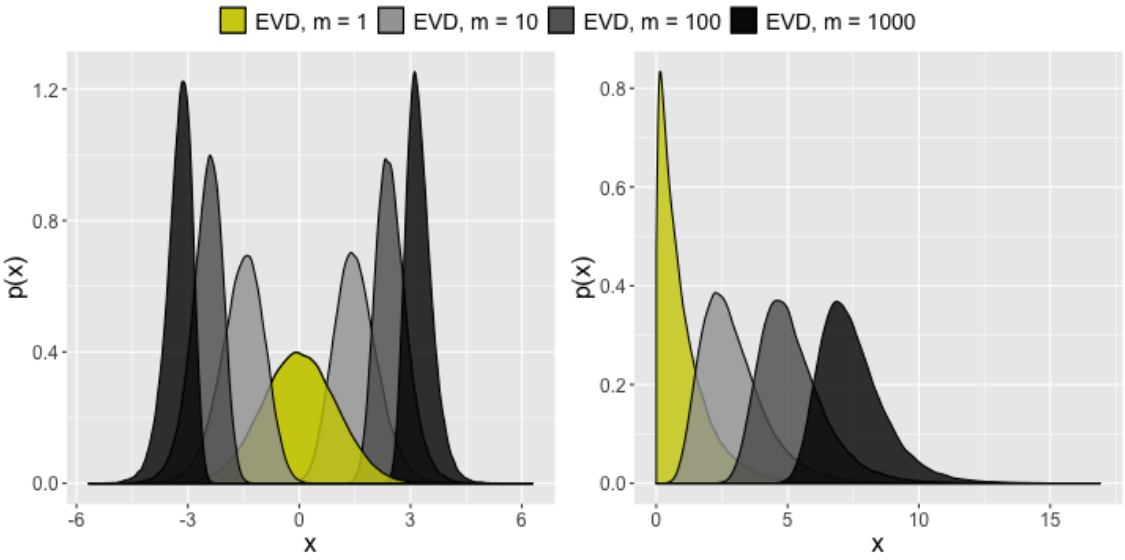

Figure 1.1: *Empirical distributions of* 10^6 *minima and maxima for the standard Gaussian distribution (left), and of maxima for the standard exponential distribution (right). (Reproduced from Hugueny, 2013, p.87.)*

Theorem 1.1 (Fisher-Tippett theorem, limit laws for maxima)**.** *(Theorem 3.2.3 in* Embrechts, Klüppelberg, and Mikosch (2013)*, p. 121; the notations have been changed for consistency within this thesis.)*

Let $\boldsymbol{X} = \{x_1, x_2, \ldots, x_m\}$ *be a sequence of iid random variables and* $X_{max} = max(\boldsymbol{X})$*. If there exists a centring constant* $d_m (\in \Re)$ *and normalising constant* $c_m (> 0)$*, and some non-degenerate distribution function* H^+ *('+' refers to the distribution of maxima) such that:*

$$c_m^{-1}(X_{max} - d_m) \xrightarrow{d} H^+,$$

then H^+ *belongs to one of the following three distribution function types:*

$$\textit{Fréchet}: \quad \Phi_\alpha^+(x) = \begin{cases} 0, & x \leqslant 0 \\ exp\{-x^{-\alpha}\}, & x > 0 \end{cases} \qquad \alpha > 0$$

$$\textit{Weibull}: \quad \Psi_\alpha^+(x) = \begin{cases} exp\{-(-x)^{\alpha}\}, & x \leqslant 0 \\ 1, & x > 0 \end{cases} \qquad \alpha > 0$$

$$\textit{Gumbel}: \quad \Lambda^+(x) = exp\{-e^{-x}\}, \qquad x \in \Re.$$

Definition 1.1 (Extreme value distribution and extremal random variable)**.** (Definition 3.2.6 in Embrechts, Klüppelberg, and Mikosch (2013), p. 124)

The distribution functions $\Phi_\alpha, \Psi\alpha$ and Λ as presented in Theorem 1.1 are called standard extreme value distributions and the corresponding random variables, standard extremal random variables. Distribution functions of the types of $\Phi_\alpha, \Psi\alpha$ and Λ are extreme value distributions; the corresponding random variables are extremal random variables.

□

From Theorem 1.1, it can be observed that the extreme value distributions are implicitly parameterised by m, the size of the sample from which the extrema is taken. Therefore, different values of m will yield different extreme value distributions (Clifton, Hugueny, and Tarassenko, 2011).

Definition 1.2 (Maximum domain of attraction)**.** (Definition 3.3.1 in Embrechts, Klüppelberg, and Mikosch (2013), p. 128; the notations have been changed for consistency within this thesis.)

We say that the rv X (the distribution function F of X or the distribution of X) belongs to the maximum domain of attraction of the extreme value distribution H^+ if there exist constants $c_n > 0$, $d_n \in \Re$ such that:

$$c_m^{-1}(X_{max} - d_m) \xrightarrow{d} H^+.$$

We write $X \in MDA(H^+)$ $(F \in MDA(H^+))$.

□

The following properties, highlighted by Embrechts, Klüppelberg, and Mikosch (2013), will assist in deciding the maximum domain of attraction of the three extreme value distributions to which X belongs. Let $x_F = \sup\{x \in \Re : F(x) < 1\}$ denote the right endpoint of F.

- All distribution functions $F \in MDA(\Phi_\alpha^+)$ have an infinite right endpoint $x_F = \infty$ (the tail decreases like a power low). The Pareto, F, Cauchy and log-gamma distribution

functions are just a few examples covered by the maximum domain of attraction of the Fréchet distribution.

- All distribution functions $F \in MDA(\Psi_\alpha^+)$ have a finite right endpoint $x_F < \infty$ (truncated tail). The uniform and beta distributions are two examples covered by the maximum domain of attraction of the Weibull distribution.

- Unlike the Fréchet and Weibull distributions, the maximum domain of attraction of the Gumbel distribution is not easy to characterise, because all distribution functions $F \in MDA(\Lambda^+)$ can have either a finite or an infinite endpoint $x_F \leqslant \infty$. Perhaps one way of thinking of the maximum domain of attraction of the Gumbel distribution is that it consists of distribution functions whose right tail decreases to zero faster than any power function (exponentially decaying tail). The exponential, gamma, normal and lognormal distributions are just a few examples covered by the maximum domain of attraction of the Gumbel distribution.

Extreme value distributions for minima can be discussed in a similar manner. In Chapter 3, we are particularly interested in the Weibull extreme value distribution for minima, which is given by

$$\Psi_\alpha^-(x) = \begin{cases} 0, & x < 0 \\ 1 - exp\{-x^{-\alpha}\}, & x \geqslant 0 \end{cases}$$

where '-' refers to the distribution of minima.

Interested readers are referred to the work of Embrechts, Klüppelberg, and Mikosch (2013) for a detailed discussion of the characterisation of the three classes: Gumbel, Fréchet and Weibull.

Definition 1.3 (Quantile function)**.** (Definition 3.3.5 in Embrechts, Klüppelberg, and Mikosch (2013), p.130)

The generalised inverse of the distribution function F

$$F^{\leftarrow}(t) = \inf\{x \in \Re : F(x) \geqslant t\}, \quad 0 < t < 1,$$

is called the quantile function of the distribution function F. The quantity $x_t = F^{\leftarrow}(t)$ defines the t-quantile of F.

□

Theorem 1.2 (Maximum domain of attraction of Ψ_α^-)**.** *(Theorem 1 in Clifton, Hugueny, and Tarassenko (2011), p. 384; the notations have been changed for consistency within this thesis)*

The distribution function F belongs to the maximum domain of attraction of the minimal Weibull distribution (Ψ_α^-), $\alpha > 0$, if and only if $x_F > -\infty$ and $F(x_F + x^{-1}) = x^{-\alpha}L(x)$ for some slowly varying function L.

If $F \in MDA(\Psi_\alpha^-)$, then

$$c_m^{-1}(X_{min} - x_F) \underrightarrow{d} \Psi_\alpha^-,$$

where the normalising constant c_m and the centring constant d_m can be chosen as $c_m = x_F + F^{\leftarrow}(m^{-1})$ and $d_m = x_F$. X_{min} is the minimum of m data. $x_F = inf\{x \in \Re : F(x) \leqslant 0\}$. $F^{\leftarrow}(t)$ is the t-quantile of F. L is a slowly varying function at ∞; that is, a positive function for all $t > 0$ that obeys

$$lim_{x \longrightarrow \infty} \frac{L(tx)}{L(x)} = 1.$$

Although these extreme value distributions differ in their purposes for modelling, they are related closely from a mathematical point of view. The following properties can be verified immediately (Embrechts, Klüppelberg, and Mikosch, 2013; Hugueny, 2013):

$$\begin{aligned} & X^{-1} \in MDA(\Psi_\alpha^-) \text{ with shape parameter } \alpha \\ \iff \quad & -X^{-1} \in MDA(\Psi_\alpha^+) \text{ with shape parameter } \alpha \\ \iff \quad & ln(X)^\alpha \in MDA(\Lambda^+). \end{aligned}$$

□

Let $X_1, X_2, \ldots, X_n$ be a sample from a distribution function F and let $X_{1:n} \geqslant X_{2:n} \geqslant \cdots \geqslant X_{n:n}$ be the order statistics. The available data are $X_{1:n}, \ldots, X_{k:n}$ for some fixed k.

Theorem 1.3 (Spacing theorem)**.** *(Proposition 1 in Burridge and Taylor (2006), p. 6 and Theorem 3 in Weissman (1978), p. 813; the notations have been changed for consistency in this thesis.)*

Let $D_{i,n} = X_{i:n} - X_{i+1:n}$, $(i = 1, \ldots, k)$ be the spacing between successive order statistics. If F is in the maximum domain of attraction of the Gumbel distribution, then spacings $D_{i,n}$ are asymptotically independent and exponentially distributed with mean proportional to i^{-1}.

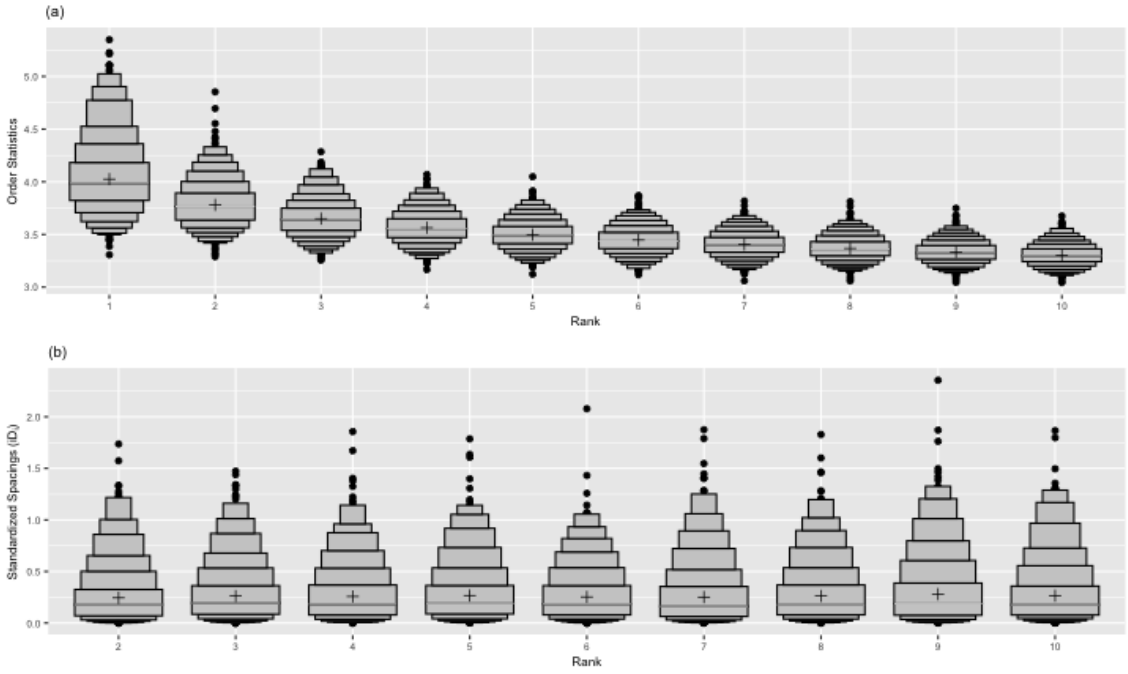

Figure 1.2: *(a) Distribution of the descending order statistics $X_{i:n}$ and (b) distribution of the standardised spacings $iD_{i,n}$ for $i \in \{1, \ldots, 10\}$ for $1,000$ samples each containing $20,000$ random numbers from the standard normal distribution.*

This theorem is illustrated using Figure 1.2, which shows the distribution of the descending order statistics $(X_{i:n})$ and the standardized spacings, $(iD_{i,n})$, for $i \in \{1, \ldots, 10\}$ for $1,000$ samples each containing $20,000$ random numbers from the standard normal distribution. Figure 1.2 (a) shows the distribution of $X_{i:n}$ with means of $X_{i:n}$ depicted as black crosses. The gaps between consecutive black crosses give the spacings between higher-order statistics $(D_{i,n})$. We note that the normal distribution is in the maximum domain of attraction of

the Gumbel distribution and that this example contains no outliers. A consequence of Theorem 1.3 is that the standardised spacings $(iD_{i,n})$ for $(i = 1, \ldots, K)$, are approximately iid (Burridge and Taylor, 2006). Figure 1.2 (b) shows the distribution of the standardised spacings $(iD_{i,n})$ for $(i = 1, 2, \ldots, 10)$ for $1,000$ samples of size $20,000$. Each letter-value box plot (Hofmann, Wickham, and Kafadar, 2017) exhibits approximately the shape of an exponential distribution.

Definition 1.4 (Distribution of probability densities)**.** (Definition 6 in Hugueny (2013), p. 105)

Let X be a (possibly multivariate) random variable with CDF F, pdf f, support D_f, and probability space P_f. $Y = f(X)$ is a random variable, distributed according to:

$$\begin{aligned} \forall y \in P_f, \quad G(y; f) &= P(Y \leqslant y) \\ &= P(f(X) \leqslant y) \\ &= \int_{f^{-1}(]y_{min}, y])} f(x)dx, \end{aligned}$$

where $y_{min} = \mathrm{Inf}(P_f)$ and $f^{-1}(]y_{min}, y])$ is the preimage of $]y_{min}, y]$ under f.

□

Proposition 1.1 (The domain of attraction of the distribution of probability density (DPD) for the multivariate Gaussian)

(Proposition 4 in Hugueny (2013), p. 141; *the notations have been changed for consistency within this thesis.*)

Let f be the multivariate Gaussian distribution. F is for the distribution of its minima in probability density in the domain of attraction of the minimum Weibull distribution Ψ_α^-.

For a sample size $m \in N^*$, the norming constants can be chosen to be:

$$\begin{aligned} c_m &= G^{\leftarrow}(\tfrac{1}{m}; f) \\ d_m &= 0 \\ \alpha_m &= 1. \end{aligned}$$

1.1.5 Calculation of Anomalous Threshold

Chapter 2 and Chapter 4 both use Theorem 1.3 (Spacing Theorem by Weissman (1978)) to estimate a data-driven anomalous threshold to discriminate anomalies. This step sorts anomalous scores and searches for any large gap at the upper tail of the distribution defined by the anomalous scores. This search for significant gaps in the upper tail can either be performed using the top-down algorithm by Burridge and Taylor (2006) or bottom-up algorithm by Schwarz (2008).

Top-down algorithm

As the name implies, the top-down algorithm introduced by Burridge and Taylor (2006) starts from the maximum and moves backwards over the sorted array, seeking a significantly large gap. As summarised by Schwarz (2008), the top-down algorithm is as follows:

- Let $X_1, X_2, \ldots X_n$ be a sample from a distribution function F, and let $X_{1:n} \geqslant X_{2:n} \geqslant \cdots \geqslant X_{n:n}$ be the order statistics.
- Let $D_i = X_{i:n} - X_{i+1:n}$ be the spacing between successive order statistics.
- Calculate the standardised spacings, $S_i \equiv iD_i$.
- Find the maximum of the first N/α spacings, S_k, where N is the maximum possible number of outliers and α is the acceptable false positive rate. The quantity N/α then represents the number of spacings that must be examined to achieve a significance level of α.
- If $k \leq N$ spacings, mark the top k values as anomalies.
- In addition to the gap between anomalous points and the valid data, sometimes there can be multiple gaps in between different groups of anomalous points. Therefore, repeat the above steps on the remaining data until no more gaps are found in the top N values.

However, repeating the process over data until it detects all the discrete groups of anomalies present in the dataset makes the algorithm inefficient for massive datasets with vast quantum of data. Further, it is not desirable to set a value for the maximum possible number of outliers, because this value is not known in advance for many real-world applications.

Ideally, the algorithm should be able to pick all the anomalies present in the data without having this predetermined number. Further, according to Schwarz (2008), this algorithm does not use the full power of the spacing theorem; it only employs the fact that the standardised spacings are iid and fails to use the fact that they are exponentially distributed, which could have given more information about how unlikely is a given spacing. The bottom-up algorithm introduced by Schwarz (2008) has the ability to release these unrealistic assumptions and overcome the limitations of the top-down algorithm. The bottom-up algorithm is based on the work of Burridge and Taylor (2006) but uses the full power of the spacing theorem.

Bottom-up algorithm

As in the top-down algorithm, the bottom-up algorithm is also based on the assumption that anomalies can bring large separations between valid data and anomalies, compared with the separations between valid data among themselves. However, in contrast to the top-down algorithm, the bottom-up algorithm now starts from the middle of the sorted data array, which represents the valid data, and moves forward towards the upper tail of the sorted array until it reaches a large gap, which is highly unlikely to occur if it is generated from the same distribution of the valid data. When a gap is encountered that is well beyond expectation, it terminates the searching process and marks all the points above that value as outliers. The specific steps of the bottom-up algorithm proposed by Schwarz (2008) is as follows:

- Let $X_1, X_2, \ldots X_n$ be a sample from a distribution function F, and let $X_{1:n} \geqslant X_{2:n} \geqslant \cdots \geqslant X_{n:n}$ be the order statistics.
- Calculate $D_i = X_{i:n} - X_{i+1:n}$, the spacing between successive order statistics.
- At each rank i, test the hypothesis that $X_{i:n}$ is the largest valid data point in the sample, with the help of the spacings immediately below it $(D_{i+1}, D_{i+2}, ..)$. If $X_{i:n}$ is the largest valid data point in the sample, according to the spacing theorem, spacings $D_i, D_{i+1}, D_{i+2}, \ldots$ should be proportional to $1, \frac{1}{2}, \frac{1}{3}, \ldots$, and so on. This allows us to use spacings $D_{i+1,n}, D_{i+2,n}, \ldots, D_{i+k,n}$ to predict the spacing D_i:

$$\hat{D}_i = \frac{1}{k-1} \sum_{j=2}^{k} j D_{i+j-1}$$

(Since the spacing theorem applies only to a small fraction of the data ranked near the upper tail, the entire dataset cannot be used to estimate D_i and therefore is used only $k(\ll n)$ number of spacings for the estimation process. The value k should be large enough to obtain a stable estimate for D_i, but small compared with the sample size, n. Schwarz (2008) has recommended 50 spacings as a rough guideline for the value k and the same has been used by Wilkinson (2018) for large samples). If they all represent valid points, then all the terms in the summation have the same mean, which is similar to the mean of D_i. Therefore, $\hat{D}_i$ serves as an estimator for D_i.

- As in the spacing theorem, since D_i follows an exponential distribution with mean proportional to i^{-1}, for a given significance level α, a threshold t that will not be exceeded by valid data can be obtained using:

$$t = \hat{D}_i log(1/\alpha)$$

- Work upward towards the upper tail of the sorted data array. At the first i where spacing D_i exceeds threshold t, terminate the searching process and flag $X_{i:n}$ and all the points above in the sorted array as outliers.

The bottom-up algorithm has been used in the research presented in Chapter 2 and Chapter 4 because of its obvious advantages over the top-down algorithm.

1.2 Motivation and Objectives

In light of the increasing demand for accurate and powerful automated methods for early detection of anomalies in the streaming data scenario and the lack of attention paid to this topic, the primary motivation of this thesis is to develop methods for early detection of anomalies in the streaming data context.

The **first** motivation of this thesis arises from the recently proposed HDoutliers method by Wilkinson (2018). The HDoutliers algorithm is a powerful algorithm with a strong theoretical foundation for anomaly detection in high-dimensional data. However, some limitations significantly hinder its performance level. The effect of these limitations is a tendency to increase the rate of false positives and/or rate of false negatives under certain

conditions. Therefore, the first objective is to propose solutions to these limitations of the HDoutliers algorithm. Chapter 2 addresses this objective. The proposed algorithm, the stray algorithm, is based on distance measures and the extreme value theory. Chapter 2 also demonstrates how the stray algorithm can assist in detecting anomalies in other data structures, such as time series data and streaming temporal data. The improved algorithm is implemented in the open source R package `stray`}.

The **second** motivation of this thesis originated from the limited research attempts on detecting anomalous series within a large collection of series in the streaming data scenario where data flow rapidly in a continuous manner. A few researchers (Hyndman, Wang, and Laptev, 2015; Wilkinson, 2018) have developed methods to identify anomalous series within a large collection of series, mainly focusing on the batch scenario where it is assumed that the entire dataset is available prior to analysis. However, in contrast to the batch scenario, the streaming data scenario poses many different challenges owing to its complex nature evolving over time. In addition to the obvious difficulties caused by the large volume and velocity of streaming data, highly noisy signals can increase the related complexity. Nonstationarity (concept drift) is another major topic in the streaming data analysis that makes it difficult to distinguish new typical behaviour from anomalous events (Faria et al., 2016). To address this issue, detectors should be able to learn and adapt according to the conditions present. Early detection of anomalies as soon as they start but before they end is another major requirement of most applications related to this problem. Therefore, the second objective of this study is to develop a powerful automated method to detect anomalous series within a large collection of series in the streaming data context such that it meets these requirements. Chapter 3 is dedicated to achieving this objective. This chapter presents a new algorithm based on density modelling and the extreme value theory. To cope with nonstationarity (concept drift), a density-based comparison approach is proposed. The proposed algorithm can detect significant changes in the typical behaviour and automatically update the anomalous threshold upon detecting a nonstationarity. The proposed algorithm is implemented in the open source R package `oddstream`.

The **third** motivation of this thesis arises owing to the non-existence of a customised method to detect technical anomalies in high-frequency water-quality data from *in situ*

sensors. Automated *in situ* sensors have the potential to revolutionise the way we manage and monitor environmental settings, such as air, soil and water. The data produced by these sensors enable us to identify fine-scale patterns, trends and extremes over space and time. Although they represent cutting-edge technology, the data they produce are still prone to errors because of many reasons, such as miscalibration, biofouling and battery failures (Horsburgh et al., 2015). Moreover, these anomalies and the ability to detect them can differ according to the geographic characteristics of the environmental system and the spatial placement of the sensors. To ensure data quality, we need to automate the real-time detection of anomalies. Therefore, our third objective is to propose a new framework for automated anomaly detection in high-frequency water-quality data from *in situ* sensors. Chapters 4 and 5 address this objective. In these chapters, an attempt was made to develop methods that can incorporate the correlation structure of several measurements taken from each site. This involves an application performing anomaly detection using turbidity, conductivity and river level data collected from rivers flowing into the Great Barrier Reef lagoon, Australia. The proposed algorithm is implemented in the open source R package `oddwater`.

Conclusions are drawn in Chapter 6 with a discussion on potential extensions to the proposed algorithms introduced in Chapter 2, 3 and 4.

These three main objectives guided the structuring and development of the major chapters of this thesis. Since this is a thesis by publication that has an introductory chapter and a concluding chapter with articles in between, the reader may notice some amount of repetition among chapters. Each article should be self-contained and therefore has been published with relevant materials for completeness.

Chapter 2

Anomaly Detection for High Dimensional Data

This article has been submitted to *Computational Statistics & Data Analysis* for possible publication.

Anomaly Detection in High Dimensional Data

Priyanga Dilini Talagala
Department of Econometrics and Business Statistics, Monash University, Australia, and
ARC Centre of Excellence for Mathematics and Statistical Frontiers
Email: dilini.talagala@monash.edu
Corresponding author

Rob J. Hyndman
Department of Econometrics and Business Statistics, Monash University, Australia, and
ARC Centre of Excellence for Mathematics and Statistical Frontiers

Kate Smith-Miles
School of Mathematics and Statistics, University of Melbourne, Australia, and
ARC Centre of Excellence for Mathematics and Statistical Frontiers

15 August 2019

JEL classification: C1, C8, C55

Anomaly Detection in High Dimensional Data

Abstract

The HDoutliers algorithm is a powerful unsupervised algorithm for detecting anomalies in high-dimensional data, with a strong theoretical foundation. However, it suffers from some limitations that significantly hinder its performance level, under certain circumstances. In this article, we propose an algorithm that addresses these limitations. We define an anomaly as an observation that deviates markedly from the majority with a large distance gap. An approach based on extreme value theory is used for the anomalous threshold calculation. Using various synthetic and real datasets, we demonstrate the wide applicability and usefulness of our algorithm, which we call the `stray` algorithm. We also demonstrate how this algorithm can assist in detecting anomalies present in other data structures using feature engineering. We show the situations where the stray algorithm outperforms the HDoutliers algorithm both in accuracy and computational time. This framework is implemented in the open source R package `stray`.

Keywords: Extreme value theory, High-dimensional data, Nearest neighbour searching, Temporal data, Unsupervised outlier detection

1 Introduction

The problem of anomaly detection has many different facets, and detection techniques can be highly influenced by the way we define anomalies, type of input data and expected output. These differences lead to wide variations in problem formulations, which need to be addressed through different analytical techniques. Although several useful computational methods currently exist, developing new methods for anomaly detection continues to be an active, attractive interdisciplinary research area owing to different analytical challenges in various application fields, such as environmental monitoring (Talagala et al. 2019b; Leigh et al. 2019), object tracking (Gupta et al. 2014; Sundaram et al. 2009), epidemiological outbreaks (Gupta et al. 2014), network security (Hyndman, Wang & Laptev 2015; Cao et al. 2015) and fraud detection (Talagala et al. 2019a). Ever-increasing computing resources and advanced data collection technologies that

emphasise real-time, large-scale data are other reasons for this growth since they introduce new analytical challenges with their increasing size, speed and complexity that demand effective, efficient analytical and computing techniques.

Anomaly detection has two main objectives, which are conflicting in nature: One downgrades the value of anomalies and attempts eliminating them, while the other demands special attention be paid to anomalies and root-cause analysis be conducted. The presence of anomalies in data can be considered data flaws or measurement errors that can lead to biased parameter estimation, model misspecification and misleading results if classical analysis techniques are blindly applied (Ben-Gal 2005; Abuzaid, Hussin & Mohamed 2013). In such situations, the focus is to find opportunities to remove anomalous points and thereby improve both the quality of the data and results from the subsequent data analysis (Novotny & Hauser 2006). In contrast, in many other applications, anomalies themselves are the main carriers of significant and often critical information, such as extreme weather conditions (e.g., bushfire, tsunami, flood, earthquake, volcanic eruption and water contamination), faults and malfunctions (e.g., flight tracking and power cable tracking) and fraud activities (Ben-Gal 2005), that can cause significant harm to valuable lives and assets if not detected and treated quickly.

High-dimensional datasets exist across numerous fields of study (Liu et al. 2016). Some anomaly detection algorithms also use feature engineering as a dimension reduction technique and thereby convert other data structures, such as a collection of time series using time series features (Talagala et al. 2019b; Hyndman, Wang & Laptev 2015), collection of scatterplots using scagnostics (Wilkinson, Anand & Grossman 2005) and genomic micro arrays and chemical compositions in biology (Liu et al. 2016) into high-dimensional data prior to the detection process for easy control. Under the high-dimensional data scenario, all attributes can be of the same data type or a mixture of different data types, such as categorical or numerical, which has a direct impact on the implementation and scope of the algorithm. Much research attention has been paid to anomaly detection for numerical data (Breunig et al. 2000; Tang et al. 2002; Jin et al. 2006; Gao et al. 2011). Limited methods are available that treat both numerical and categorical data using correspondence analysis, for example, as in Wilkinson (2017).

High-dimensional anomalies can arise in all the attributes or a subset of the attributes (Unwin 2019). If all anomalies in a high-dimensional data space were anomalies in a lower dimension, then anomaly detection can be performed using axis parallel views or by incorporating an additional step of variable selection for the detection process. However, in practice, certain high-dimensional instances are only perceptible as anomalies if treated as high-dimensional

problems and the correlation structure of all the attributes considered. Otherwise, these tend to be overlooked if attributes are considered separately (Wilkinson 2017; Ben-Gal 2005).

The problem of anomaly detection has been extensively studied over the past decades in many application domains. Several surveys of anomaly detection techniques have been conducted in general (Chandola, Banerjee & Kumar 2009; Aggarwal 2017) or for specific data domains such as high-dimensional data, network data (Shahid, Naqvi & Qaisar 2015), temporal data (Gupta et al. 2014), machine learning and statistical domains (Hodge & Austin 2004), novelty detection (Pimentel et al. 2014), intrusion detection (Sabahi & Movaghar 2008) and uncertain data (Aggarwal & Yu 2008). Some algorithms are application specific and take advantage of the underlying data structure or other domain-specific knowledge (Talagala et al. 2019b). More general algorithms without domain-specific knowledge are also available with their own strengths and limitations (Breunig et al. 2000; Tang et al. 2002; Jin et al. 2006; Gao et al. 2011). Among the many possibilities, the HDoutliers algorithm, recently proposed by Wilkinson (2017), is a powerful unsupervised algorithm, with a strong theoretical foundation, for detecting anomalies in high-dimensional data. Although this algorithm has many advantages, a few characteristics hinder its performance. In particular, under certain circumstances it tends to increase the rate of false negatives (i.e., the detector ignores points that appear to be real anomalies) because it uses only the nearest-neighbour distances to distinguish anomalies. Further, to deal with large datasets with numerous observations it uses the Leader algorithm (Hartigan & Hartigan 1975), which forms several clusters of points in one pass through the dataset using a ball of a fixed radius. By incorporating this clustering method, it tries to gain the ability to identify anomalous clusters of points. However, in the presence of very close neighbouring anomalous clusters it tends to increase the rate of false negatives. Further, this additional step of clustering has a serious negative impact on the computational efficiency of the algorithm when dealing with large datasets.

Through this study, we make three fundamental contributions. First, we propose an algorithm called *stray*, representing '**S**earch and **TR**ace **A**nomal**Y**', that addresses the limitations of the HDoutliers algorithm. The stray algorithm presented here focuses specifically on fast, accurate anomalous score calculation using simple but effective techniques for improved performance. Second, we introduce an R (R Core Team 2019) package, `stray` (Talagala, Hyndman & Smith-Miles 2019), that implements the stray algorithm and related functions. Third, we demonstrate the wide applicability and usefulness of our stray algorithm, using various datasets.

Our improved algorithm, stray, has many advantages: (1) It can be applied to both one-dimensional and high-dimensional data. (2) It is unsupervised in nature and therefore does not require training datasets for the model-building process. (3) The anomalous threshold is a data-driven threshold and has a valid probabilistic interpretation because it is based on the extreme value theory. (4) By using k-nearest neighbour distances for anomalous score calculation, it gains the ability to deal with the masking problem. (5) It can provide near real-time support to datasets that stream in large quantities owing to its use of fast nearest neighbour searching mechanisms. (6) It can deal with data that may have multimodal distributions for typical data instances. (7) It produces both score (to indicate how anomalous the instances are) and binary classification (to reduce the searching space during the visual and root-cause analysis) for each data instance as an output. (8) It can detect outliers as well as inliers.

The remainder of this paper is organised as follows. Section 2 presents the related work to lay the foundation for the stray algorithm. Section 3 describes the limitations of the HDoutliers algorithm that hinder its performance. Section 4 presents the improved algorithm, stray, that addresses the limitations of the HDoutliers algorithm. Section 5 presents a comprehensive evaluation, illustrating the key features of the stray algorithm. Section 6 includes an application of stray algorithm related to pedestrian behaviour in the city of Melbourne, Australia. Section 7 concludes the article and presents future research directions.

2 Background

2.1 Types of Anomalies in High Dimensional Data

The problems of anomaly detection in high-dimensional data are threefold (Figure 1), involving detection of: (a) global anomalies, (b) local anomalies and (c) micro clusters or clusters of anomalies (Goldstein & Uchida 2016). Most of the existing anomaly detection methods for high-dimensional data can easily recognise global anomalies since they are very different from the dense area with respect to their attributes. In contrast, a local anomaly is only an anomaly when it is distinct from, and compared with, its local neighbourhood. Madsen (2018) introduces a set of algorithms based on a density or distance definition of an anomaly, which mainly focuses on local anomalies in high-dimensional data. Micro clusters or clusters of anomalies may cause masking problems. Very little attention has been paid to this problem relative to the other two categories. The recently proposed HDoutliers algorithm (Wilkinson 2017) addresses this problem to some extent by grouping instances together that are very close in the

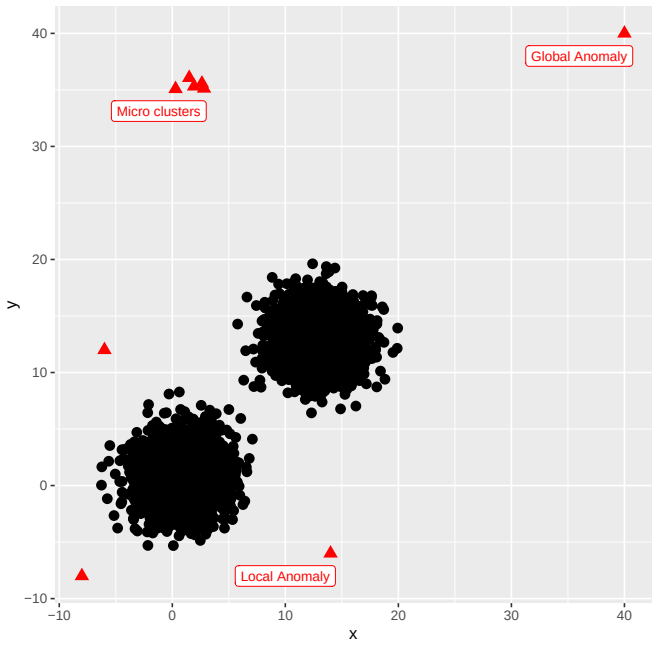

Figure 1: *Different types of anomalies in high-dimensional data. Anomalies are represented by red triangles and black dots correspond to the typical behaviour.*

high-dimensional space and then selecting a representative member from each cluster before calculating nearest neighbour distances for the selected instances. In this study, we focus on all three of these anomaly types.

2.2 Definitions for Anomalies in High Dimensional Data

Anomalies are often mentioned in the literature under several alternative terms, such as outliers, novelty, faults, deviants, discordant observations, extreme values/cases, change points, rare events, intrusions, misuses, exceptions, aberrations, surprises, peculiarities, odd values and contaminants, in different application domains (Chandola, Banerjee & Kumar 2009; Gupta et al. 2014; Zhang, Wu & Yu 2010). Of these, the two terms anomalies and outliers are used commonly and interchangeably in the literature describing research related to the topic. The term inlier also relates to the topic, but rarely appears in the literature on anomaly detection. Inliers are those points that appear between typical clusters without attaching to any of the clusters, but still lie within the range defined by the typical clusters (Jouan-Rimbaud et al. 1999). In contrast, the corresponding notion of an 'outlier' is generally used to refer to a data instance that appears out of the space more towards the tail of a distribution, defined by the typical data instances. Some classical methods related to the topic fail to detect inliers and only focus on outliers (Jouan-Rimbaud et al. 1999). However, detecting inliers is equally important because they can give rise to interpolation errors. In this study, we focus on both inliers and outliers. To

avoid any confusion, we use the term 'anomaly' for the purpose of nomenclature throughout this paper.

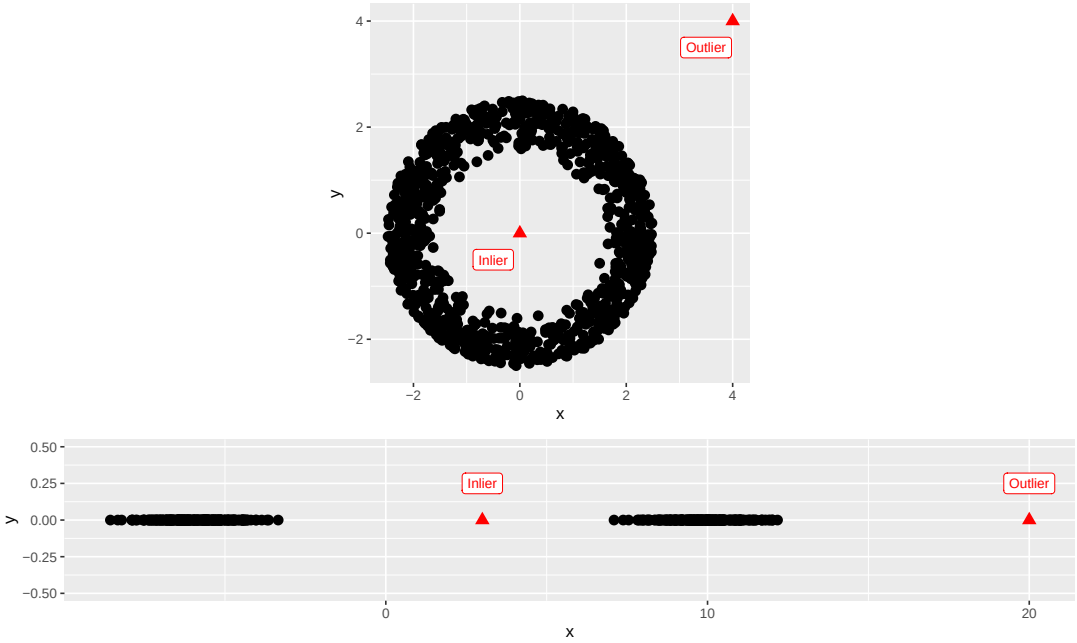

Figure 2: *Inliers vs outliers. Anomalies are represented by red triangles and black dots correspond to the typical behaviour.*

Owing to the complex nature of the problem, it is difficult to find a unified definition for an anomaly and the definition often depends on the focus of the study and the structure of the input data available to the system (Williams 2016; Unwin 2019). However, there are some definitions that are general enough to cope with datasets with various application domains. Grubbs (1969) defines an anomaly as an observation that deviates markedly from other members of the dataset. However, this deviation can be defined in terms of either distance or density. Burridge & Taylor (2006), Wilkinson (2017) and Schwarz (2008) have all proposed methods for anomaly detection by defining an anomaly in terms of distance. In contrast, Hyndman (1996), Clifton, Hugueny & Tarassenko (2011) and Talagala et al. (2019a) have proposed methods that define an anomaly with respect to either the density or the chance of the occurrence of observations. Madsen (2018) also provides a series of distance and density-based anomaly detection algorithms.

In this study, we define an anomaly as an observation that deviates markedly from the majority with a large distance gap under the assumption that there is a large distance between typical data and the anomalies compared with the distance between typical data.

3 Limitations of HDoutliers Algorithm

Although the HDoutliers algorithm (Wilkinson 2017) has many advantages, a few characteristics limit its possibilities. Next, we discuss these limitations in detail.

3.1 HDoutliers Uses Only the Nearest Neighbour Distance to Discriminate Anomalies

The HDoutliers algorithm uses the Leader algorithm (Hartigan & Hartigan 1975) to form small clusters of points, prior to calculating nearest neighbour distance. In the Leader algorithm, each cluster is a ball in the high-dimensional data space. In the HDoutliers algorithm, the radius of this ball is selected such that it is well below the expected value of the distances between $n(n-1)/2$ pairs of points distributed randomly in a d-dimensional unit hypercube.

After forming clusters using the Leader algorithm, the HDoutliers algorithm selects representative members from each cluster. It then calculates the nearest neighbour distances for each of these representative members. These distances are then used to identify the anomalies based on the assumption that anomalies bring large distance separations between typical data and the anomalies, in comparison to the separations between typical data themselves. Therefore, under this assumption it is believed that any anomalous cluster will appear far away from the clusters of the typical data points. As a result, the nearest neighbour distance for this anomalous cluster will be significantly higher than that of the clusters of typical data and thereby identify it as an anomalous cluster. All the data points contained in the anomalous cluster are then marked as anomalous points within a given dataset.

However, one further assumption for this method to work properly is that any anomalous clusters present in the dataset are isolated. For example, imagine a situation in which two anomalous clusters are very close to one another but are far away from the rest of the typical clusters. Now, the two clusters will become nearest neighbours to one another and they will jointly project them by being anomalous by giving very small nearest neighbour distances for both clusters that are compatible with the nearest neighbour distances of the rest of the typical clusters. Figures 6 (c-II) and (d-II) further elaborate this argument. In these two examples, the HDoutliers algorithm (with the clustering step) declares points as anomalies only if they are isolated and fails to detect anomalous clusters that share a few cluster neighbours. Although the HDoutliers algorithm incorporates the clustering step with the aim of identifying anomalous clusters of points, because of the very small size of the ball that is used to produce clusters

(exemplars) in the d-dimensional space, it fails to bring all the points into a single cluster and instead produces a few anomalous clusters that are very close to one another. These anomalous clusters then become nearest neighbours to one another and have very small nearest neighbour distances for the representative member of each cluster. Since the detection of anomalies entirely depends on these nearest neighbour distances and since the anomalous clusters do not show any significant deviation from typical clusters with respect to the nearest neighbour distances, the algorithm now fails to detect these points as anomalies and thereby increases the rate of false negatives.

3.2 Problems Due to Clustering Via Leader Algorithm

After forming clusters of data points, the HDoutliers algorithm completely ignores the density of the data points. Once it forms clusters of data points using the Leader algorithm, it selects a representative member from each cluster and carries out further analysis only using these representative members. Figure 6 (e-II) provides an example related to this issue. This dataset is a bimodal dataset with an anomalous point located between the two typical classes. The entire dataset contains 2,001 data points. The data points gathered at the leftmost upper corner represent one typical class with 1,000 data points. The second typical class of data points is gathered at the rightmost bottom corner with another 1,000 data points. Since this second class of data points is closely compacted in substance, the 1,000 data points are now wrapped by a single ball when forming clusters using the Leader algorithm. In the HDoutliers algorithm, the next step is to select one member from each of these clusters. Once it selects a representative member from this ball that contains 1,000 data points, it ignores the remaining 999 data points in detecting anomalies. This step misleads the algorithm, and the remaining steps of the algorithm view this representative member as an isolated data point, although it is surrounded by 999 neighbouring data points in the original dataset. Therefore, all data points in this entire class are declared as anomalies by the algorithm, although it contains half of the dataset. Unwin (2019) suggests jittering not as a perfect solution, but as an alternative to mitigate this problem. Unwin (2019) also argues that the problem tends not to occur in high-dimensional data spaces where this kind of granularity is less likely. However, then it gives rise to the problem of neighbouring anomalous clusters (as illustrated in Figure 6 (c-II, d-II)), which individually appear to be typical, or of limited suspicion (due to the presence of other neighbouring anomalous clusters), yet, their co-occurrence is highly anomalous.

Figure 6 (f-II) provides another situation in which false negatives increase because of the clustering step. This bivariate dataset contains 1,001 data points. The data points gathered at

the leftmost upper corner represent a typical class covering 1,000 data points, and the isolated data point at the rightmost bottom corner represents an anomaly. Since this typical class of 1,000 data points is closely compacted, it gives rise to only 14 clusters through the Leader algorithm. Altogether, the dataset forms 15 clusters with the one created by the isolated point located at the rightmost bottom corner. Even though the original dataset contains 1,001 data points, the algorithm considers only 15 data points (a representative member from each cluster) for calculating the anomalous threshold. Now, this number is not large enough to yield a stable estimate for the anomalous threshold. Due to this ignorance of the density of the original dataset, it now fails to detect the obvious anomalous point at the leftmost bottom corner.

3.3 Problem with Threshold Calculation

A companion R package (Fraley 2018) is available for the algorithm proposed by Wilkinson (2017). According to the R package implementation, the current version of the HDoutliers algorithm uses the next potential candidate for anomalies in calculating the anomalous threshold, in each iteration of the bottom-up searching algorithm. This approach causes an increase in the false detection rate under certain circumstances. We avoid this limitation in our proposed algorithm.

4 Proposed Improved Algorithm: `stray` Algorithm

In this section, we propose an improved algorithm for anomaly detection in high dimensional data. Our proposed algorithm is intended to overcome the limitations of the HDoutliers algorithm and thereby enhance its capabilities.

4.1 Input to the `stray` Algorithm

An input to the stray algorithm is a collection of data instances where each data instance can be a realisation of only one attribute or a set of attributes (also referred to using terms such as features, measurements and dimensions). In this study, we limit our discussion to quantitative data; therefore, an input can be a vector, matrix or data frame of $d(\geq 1)$ numerical variables, where each column corresponds to an attribute and each row corresponds to an observation of these attributes. The focus is then to detect anomalous instances (rows) in the dataset.

4.2 Normalise the Columns

Since the stray algorithm is based on the distance definition of an anomaly, nearest neighbour distances between data instances in the high-dimensional data space are the key information for the algorithm to detect anomalies. However, variables with large variance can exert disproportional influence on Euclidean distance calculations (Wilkinson 2017). To make the variables of equivalent weight, the columns of the `data` are first normalised such that the data are bounded by the unit hypercube. This normalisation is commonly referred as *min-max normalisation*, which involves a linear transformation of the original data, with the result data ranging from 0 to 1. This type of transformation does not change the distribution or squeeze points together masking anomalies.

4.3 Nearest Neighbour Searching

In the `stray` algorithm, after the columns of the dataset are normalised, it calculates the k-nearest neighbour distance with the maximum gap for each and every instance. By using this measure, we were able to address the aforementioned limitations of the HDoutliers algorithm.

For each individual observation, the algorithm first calculates the k-nearest neighbour distances, $d_{i,KNN}$, where $i = 1, 2, ..., k$. Then, it calculates the successive differences between distances, $\Delta_{i,KNN}$. Next, it selects the k-nearest neighbour distance with the maximum gap, $\Delta_{i,max}$. Figure 3 illustrates how these steps help our improved algorithm to detect anomalous points or anomalous clusters of points.

In Figure 3 (a), the dataset contains only one anomaly at $(15, 16.5)$. For this dataset, the nearest neighbour distance can differentiate the anomalous point from the remaining typical points because the nearest neighbour distance for the anomalous point is significantly larger (14.8) than that for the remaining typical points. Figure 3 (b) shows the change in the k-nearest neighbour distances of the anomaly at $(15, 16.5)$. For this dataset, the k-nearest neighbour distance with the maximum gap occurs when $k = 1$. The second dataset, in Figure 3 b), has three anomalies around $(15, 16.5)$. If we calculate only the nearest neighbour distances for each observation, then the three anomalies are not distinguishable from the typical points since their values are very small (0.7) compared with that of most typical points with nearest neighbour distances at around $(0.0015$ to $2.5)$. However, the three anomalies are distinguishable from their typical points with respect to the k-nearest neighbour distances with the maximum gap (Figure 3 (d)). For the three anomalies in Figure 3 (d), the third nearest neighbour distance has the maximum gap (Figure 3 e)) and the three points are now easily distinguished as anomalies, with respect to

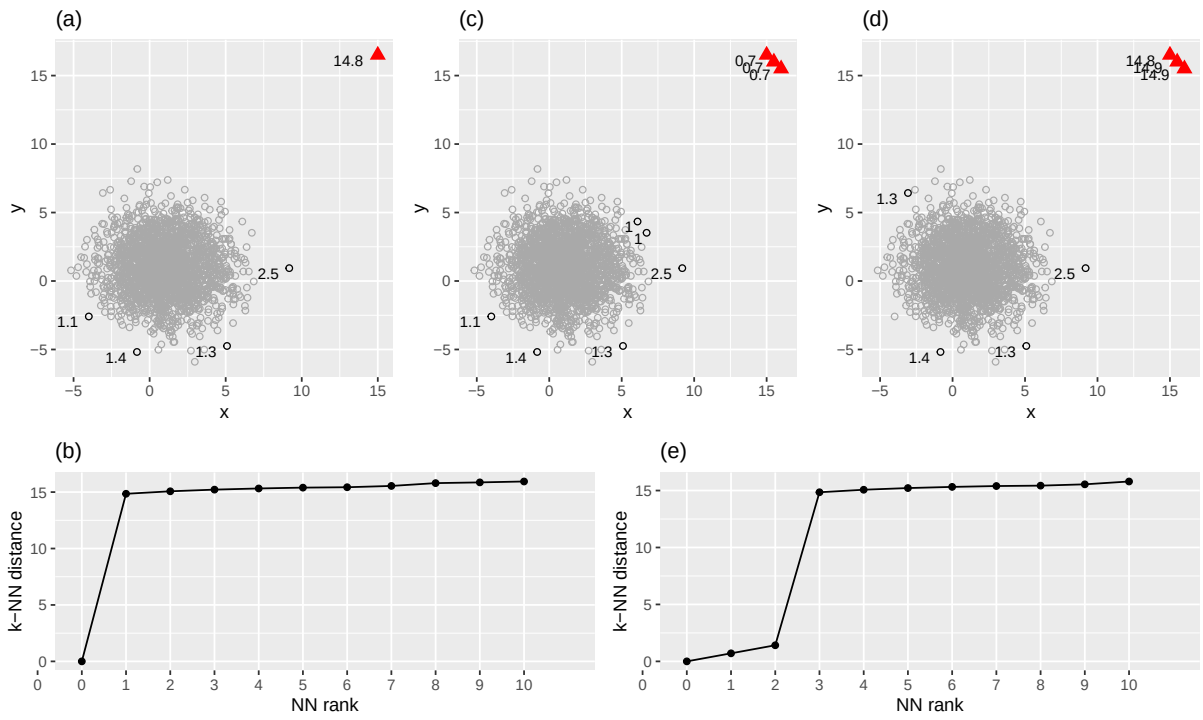

Figure 3: *Difference between the nearest neighbour distance and the k-nearest neighbour distance with the maximum gap. (a) Dataset contains only one anomaly at* $(15, 16.5)$*. Nearest neighbour distances are marked. (b) Change in the k-nearest neighbour distances of the anomaly. (c) Dataset contains micro cluster around* $(15, 16.5)$*. Nearest neighbour distances are marked. (d) Dataset contains micro cluster around* $(15, 16.5)$*. For the three anomalies, the third nearest neighbour distance has the maximum gap. (e) Change in the k-nearest neighbour distances of an anomaly from micro cluster around* $(15, 16.5)$*. Anomalies are represented by red triangles and black dots correspond to the typical behaviour.*

k-nearest neighbour distances with the maximum gap. Therefore, by using k-nearest neighbour distances with the maximum gap, the stray algorithm gains the ability to detect both anomalous singletons and micro clusters. Through this approach, we are able to reduce the false detection rate and thereby address the limitations of the HDoutliers algorithm, while gaining the ability to detect micro clusters. This is also a very simple, but clever, investment as compared with the time taken by the leader algorithm to form small clusters to detect micro clusters (especially for datasets with large dimensions), in the HDoutliers algorithm. Further, for each point, the corresponding k-nearest neighbour distances with the maximum gap act as an anomalous score to indicate the degree of being an anomaly.

In the current study, we consider both exact and approximate k-nearest neighbour searching techniques. Brute force search involves going through every possible paring of points to detect k-nearest neighbours for each data instance, and therefore, exact k-nearest neighbours are explored. Conversely, k-dimensional trees (k-d trees) employ spatial data structures that partition space to

allow efficient access to a specified query point (Elseberg et al. 2012b). Therefore, it involves searching approximate k-nearest neighbours around a specified query point.

In the current algorithm, parameter k, which determines the size of the neighbourhood, is introduced as a user-defined parameter that can be selected according to the application. One way to interpret the role of k in the stray algorithm is to view it as the minimum possible size for a typical cluster in a given dataset. If the size of an anomalous cluster is less than k, it will be detected as a micro cluster by the stray algorithm. The choice of k has different effects across different dimensions and sizes of data (Campos et al. 2016). We can set k to 1 if no micro clusters are present in the dataset and thereby focus on local and global anomalous points. High k values are recommended for datasets with high dimensions because of the curse of dimensionality.

4.4 Threshold Calculation

Anomalous scores assign each point a degree of being an anomaly. However, for certain applications it is also important to categorise typical and anomalous points for the subsequent root-cause analysis. Ideally, we prefer a universal threshold to unambiguously distinguish anomalous points from typical points. Following Schwarz (2008), the HDoutliers algorithm (Wilkinson 2017) defines an anomalous threshold based on extreme value theory, a branch of probability theory that relates to the behaviour of extreme order statistics in a given sample (Galambos, Lechner & Simiu 2013).

The anomalous threshold calculation in Schwarz (2008); Burridge & Taylor (2006) and Wilkinson (2017) is an application of Weissman's spacing theorem (Weissman 1978) (Theorem 4.1) that is applicable to the distribution of data covered by the maximum domain of attraction of a Gumbel distribution. This requirement is satisfied by a wide range of distributions, ranging from those with light tails to moderately heavy tails that decrease to zero faster than any power function (Embrechts, Klüppelberg & Mikosch 2013). Examples include the exponential, gamma, normal and log-normal distributions with exponentially decaying tails.

Let $X_1, X_2, \ldots, X_n$ be a sample from a distribution function F and let $X_{1:n} \geqslant X_{2:n} \geqslant \cdots \geqslant X_{n:n}$ be the order statistics. The available data are $X_{1:n}, \ldots, X_{k:n}$ for some fixed k.

Theorem 4.1 (Spacing Theorem)**.** *(Proposition 1 in Burridge & Taylor (2006), p.6 and Theorem 3 in Weissman (1978), p.813; the notations have been changed for consistency in this paper)*

Let $D_{i,n} = X_{i:n} - X_{i+1:n}$, $(i = 1, \ldots, k)$ be the spacing between successive order statistics. If F is in the maximum domain of attraction of the Gumbel distribution, the spacings $D_{i,n}$ are asymptotically independent and exponentially distributed with mean proportional to i^{-1}.

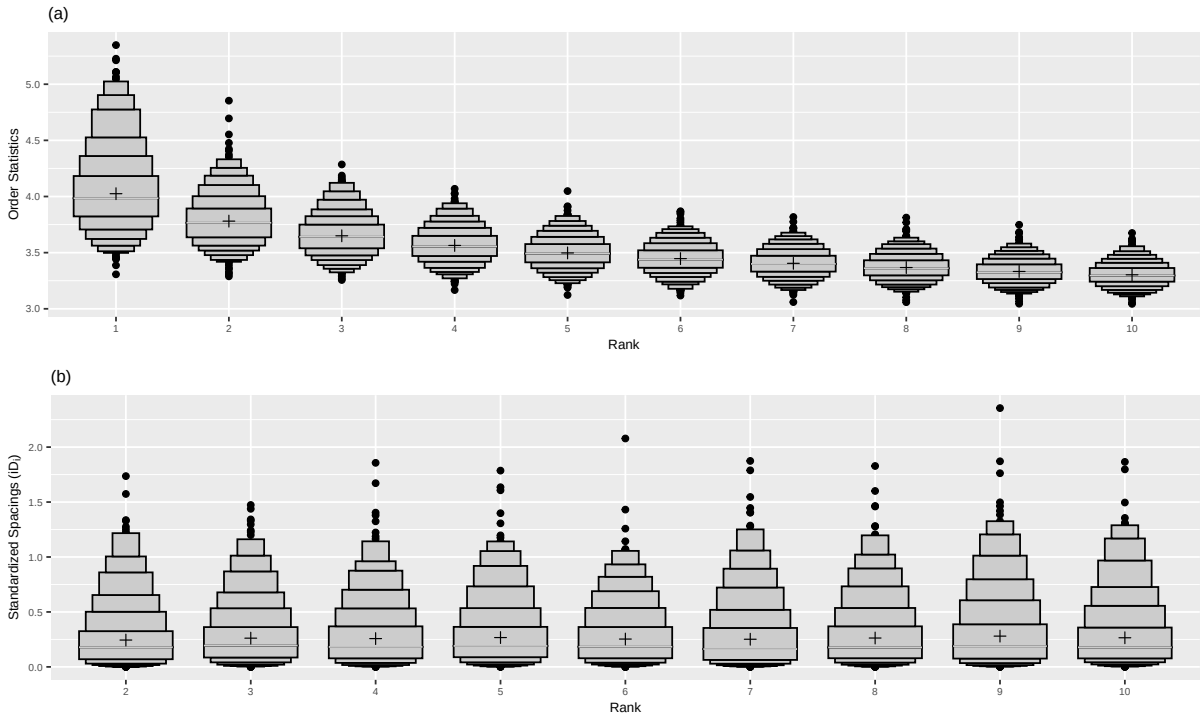

Figure 4: *(a) Distribution of the descending order statistics $X_{i:n}$ and (b) distribution of the standardised spacings $iD_{i,n}$ for $i \in \{1, \ldots, 10\}$ for $1,000$ samples each containing $20,000$ random numbers from the standard normal distribution.*

We illustrate this theorem using Figure 4, which shows the distribution of the descending order statistics $(X_{i:n})$ and the standardized spacings, $(iD_{i,n})$, for $i \in \{1, \ldots, 10\}$ for $1,000$ samples each containing $20,000$ random numbers from the standard normal distribution. Figure 4 (a) shows the distribution of $X_{i:n}$ with means of $X_{i:n}$ depicted as black crosses. The gaps between consecutive black crosses give the spacings between higher-order statistics $(D_{i,n})$. We note that the normal distribution is in the maximum domain of attraction of the Gumbel distribution and that this example contains no outliers. A consequence of Theorem 4.1 is that the standardised spacings $(iD_{i,n})$ for $(i = 1, \ldots, K)$, are approximately iid (Burridge & Taylor 2006). Figure 4 (b) shows the distribution of the standardised spacings $(iD_{i,n})$ for $(i = 1, 2, ..., 10)$ for $1,000$ samples of size $20,000$. Each letter-value box plot (Hofmann, Wickham & Kafadar 2017) exhibits approximately the shape of an exponential distribution.

Following Schwarz (2008), Burridge & Taylor (2006) and Wilkinson (2017), we start our anomalous threshold calculation from a subset of the points covering 50 per cent of them with the smallest anomalous scores under the assumption that this subset contains the anomalous scores corresponding to typical data points and the remaining subset contains the scores corresponding to the possible candidates for anomalies. Following the Weissman spacing theorem, it then fits an exponential distribution to the upper tail of the outlier scores of the first subset, and then

computes the upper $1 - \alpha$ points of the fitted cumulative distribution function, thereby defining an anomalous threshold for the next anomalous score. Then, from the remaining subset it selects the point with the smallest anomalous score. If this anomalous score exceeds the cut-off point, it flags all the points in the remaining subset as anomalies and stops searching for anomalies. Otherwise, it declares the point as a typical point and adds it to the subset of the typical points. It then updates the cut-off point, including the latest addition. This searching algorithm continues until it finds an anomalous score that exceeds the latest cut-off point. This algorithm is known as a 'bottom-up searching' algorithm in Schwarz (2008). This threshold calculation is performed under the assumption that the distribution of k-nearest neighbours with the maximum gap is in the maximum domain of attraction of the Gumbel distribution, which covers a wide range of distributions.

4.5 Output

In `stray`, anomalies are measured in two scales: (1) binary classification and (2) outlier score. Under binary classification, data instances are classified either as typical or anomalous using the data-driven anomalous threshold based on the extreme value theory. This type of classification is important if the subsequent steps of the data analysis process are automated. The stray algorithm also assigns an anomalous score to each data instance to indicate the degree of outlierness of each measurement. These anomalous scores allow the user to rank and select the most serious or relevant anomalous points for root-cause analysis and taking immediate precautions. The HDoutliers algorithm (Wilkinson 2017), which provides only a binary classification, does not directly allow the user to make such a choice to direct their attention to more significant anomalous instances. Conversely, various methods proposed in the literature provide anomalous scores, but the anomalous threshold is user defined and application specific (Madsen 2018). The output produced by `stray` is an all-in-one solution encapsulating necessary measurements of anomalies for further actions.

5 Experiments

The HDoutliers algorithm is a powerful algorithm in the current state-of-the-art methods for detecting anomalies in high-dimensional data. The focus of the stray algorithm is to address some of the limitations of the HDoutliers algorithm that hinder its performance under certain circumstances. Here, we perform an experimental evaluation on the accuracy and computational efficiency of our stray algorithm relative to the HDoutliers algorithm. While these examples are

fairly limited in number and are mostly limited to bivariate datasets, they should be viewed only as simple illustrations of the key features of the stray algorithm that outperforms the HDoutliers algorithm.

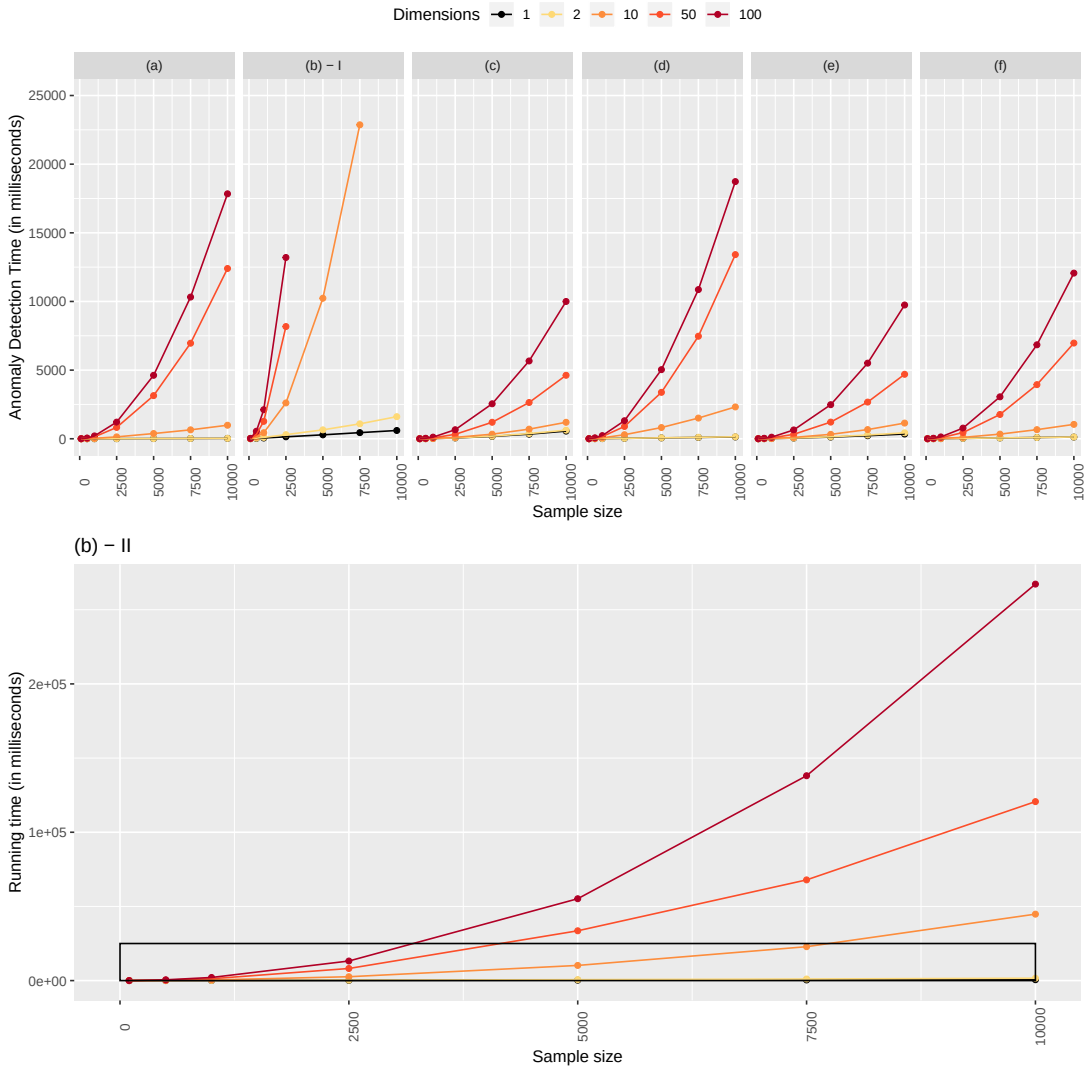

Figure 5: *Scalability Performance. (a) HDoutliers algorithm without clustering step, (b-I) HDoutliers algorithm with clustering step, (c) stray algorithm with brute force nearest neighbour search using FNN R package implementation, (d) stray algorithm with kd-trees nearest neighbour search using 'FNN' R package implementation, (e) stray algorithm with brute force nearest neighbour search using 'nabor' R package implementation, (f) stray algorithm with kd-trees nearest neighbour search using 'nabor' R package implementation. For clear comparison, only a part of the measurements of the full experiment is displayed in (b-I). (b-II) presents the full version of (b-I). Black frame in (b-II) covers the plotting region of (b-I).*

The first experiment (Figure 5) was designed to test the effect of the dimension, size of the data and the k-nearest neighbour searching method on running times of the different versions of the two algorithms: stray and HDoutliers.

The HDoutliers algorithm has two versions. The first version calculates nearest neighbour distance for each data instance and does not involve any clustering step prior to the nearest neighbour distance calculation. This version of the algorithm (version 1 of the HDoutliers, hereafter) is recommended for small samples (n<10,000). The second version uses the Leader algorithm to form several clusters of points and then selects a representative member from each cluster. The nearest neighbour distances are then calculated only for the selected representative members. Compared with version 1 of the HDoutliers algorithm (Figure 5 (a)), version 2 with the clustering step is extremely slow for higher dimensions (>10), and the running time increases more rapidly with increasing sample size. For clear comparison between the different versions of the two algorithms (stray and HDoutliers), only a part of the measurements of the full experiment of the second version of the HDoutliers algorithm is displayed in Figure 5 (b-I)). Figure 5 (b-II) presents the full version of Figure 5 (b-I). The additional clustering step in the second version of the HDoutliers algorithm, which is essential for detecting micro clusters, is extremely time-consuming, particularly with large samples with higher dimensions. Figure 5 (c)–(f) corresponds to the stray algorithm. In this experiment, to ascertain the influence from the k-nearest neighbour searching methods, we considered both exact (brute force) and approximate (kd-trees) nearest neighbour searching algorithms.

Many implementations of k-nearest neighbour searching algorithms are available for the R software environment. We considered `FNN` (Beygelzimer et al. (2019), Figure 5 (c) & (d)) and `nabor` (Elseberg et al. (2012a); Figure 5 (e) & (f)) R packages for our comparative analysis. R package `nabor`, wraps a fast k-nearest neighbour library written in templated C++. We noticed that searching $k-(>1)$ nearest neighbours (Figure 5 (a), in this example k is set to 10) instead of only one $(k=1)$ nearest neighbour (Figure 5 (d)) increases the running time only slightly as the number of instances is increased. The results in both Figure 5 (a) and Figure 5 (d) are based on approximate nearest neighbour distances using the kd-trees nearest neighbour searching algorithm. We observed that the kd-trees implementation in `nabor` package (Figure 5 (f)) is much faster than the `FFN` package implementation (Figure 5 (d)). Surprisingly, as the dimension increases, the running time of the stray algorithm with kd-trees (Figure 5 (d), (f)) increases much more quickly than that of the brute force algorithm, which involves searching every possible pairing of points to detect k-nearest neighbours for each data instance (Figure 5 (c), (e)). Other studies (Kanungo et al. 2002) have also reported a similar result for many algorithms based on kd-trees and many variants. This could be due to the parallelisability and memory access patterns of the two searching mechanisms. The brute force algorithm is easily parallelisable because it involves independent searching of all possible candidates for each data instance. In

Table 1: *Performance metrics – False positive rates. The values given are based on 100 iterations and the mean values are reported. Different versions of the two algorithms (stray and Hdoutliers) are applied on datasets where each column is randomly generated from the standardised normal distribution. All the datasets are free from anomalies HDoutliers WoC: HDoutliers algorithm without clustering step; HDoutliers WC: HDoutliers algorithm with clustering step.*

Method	dim	100	500	1000	2500	5000	7500	10000
HDoutliers WoC	1	0.017	0.011	0.008	0.007	0.005	0.005	0.004
HDoutliers WoC	10	0.002	0.002	0.002	0.002	0.002	0.002	0.002
HDoutliers WoC	100	0.001	0.001	0.001	0.001	0.001	0.001	0.001
HDoutliers WC	1	0.036	0.024	0.024	0.019	0.017	0.014	0.013
HDoutliers WC	10	0.006	0.006	0.006	0.005	0.005	0.005	0.005
HDoutliers WC	100	0.003	0.003	0.003	0.003	0.003	0.003	0.003
stray - brute force	1	0.006	0.003	0.002	0.002	0.002	0.001	0.001
stray - brute force	10	0.001	0.001	0.001	0.001	0.001	0.001	0.000
stray - brute force	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
stray - FNN kd-tree	1	0.006	0.003	0.002	0.002	0.002	0.001	0.001
stray - FNN kd-tree	10	0.001	0.001	0.001	0.001	0.001	0.001	0.000
stray - FNN kd-tree	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
stray - nabor brute	1	0.006	0.003	0.002	0.002	0.002	0.001	0.001
stray - nabor brute	10	0.001	0.001	0.001	0.001	0.001	0.001	0.000
stray - nabor brute	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
stray - nabor kd-tree	1	0.006	0.003	0.002	0.002	0.002	0.001	0.001
stray - nabor kd-tree	10	0.001	0.001	0.001	0.001	0.001	0.001	0.000
stray - nabor kd-tree	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000

contrast, the kd-tree searching algorithm is naturally serial and therefore difficult to implement on parallel systems with appreciable speedup (Zhang 2017).

Following (Wilkinson 2017), we evaluated the false positive rate (typical points incorrectly identified as anomalies) of the stray algorithm by running it many times on random data. The values presented in Table 1 are based on 1000 iterations and the mean values are reported. Different versions of the two algorithms (stray and Hdoutliers) were applied on datasets where each column is randomly generated from the standardised normal distribution. In each test, the critical value, α, was set to 0.05. Compared with the HDoutliers algorithm, low false positive rates were achieved for the stray algorithm across all dimensions and sample sizes. Unlike in the HDoutliers algorithm (Unwin 2019), in stray a much smaller false detection rate was observed even for the small datasets with smaller dimensions. No difference was observed across different versions of the stray algorithm with different nearest neighbour searching mechanisms and their different implementations.

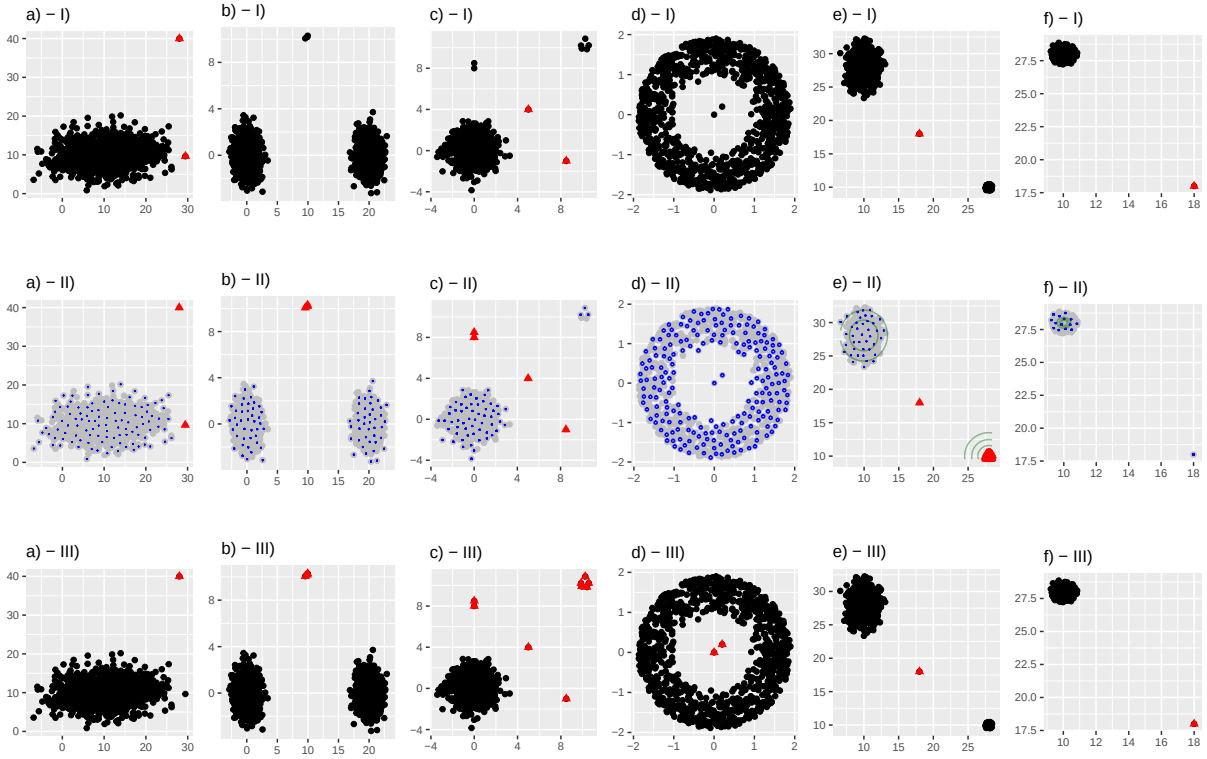

Figure 6: *Algorithm performance. (a) The top panel shows the results of the HDoutliers algorithm without a clustering step. (b) The middle panel shows the results of the HDoutliers algorithm with a clustering step. The representative member selected from each cluster formed by the Leader algorithm are marked in blue colour. (c) The bottom panel shows the results of the improved algorithm with brute force k-nearest neighbour searching. The detected anomalies are marked as red triangles.*

Figure 6 demonstrates how the stray algorithm outperforms the two versions of the HDoutliers algorithm under different circumstances. These limited set of examples were selected with the aim of highlighting some of the key feature of the stray algorithm:

(1) All three algorithms were able to correctly capture the anomalous point at the rightmost upper corner of Figure 6 (a)- I). However, the two versions of the HDoutliers algorithm tend to generate some false positives, particularly with the small dimensions.

(2) Figure 6 (b)- III) shows its ability to deal with multimodal typical classes. The two clusters at the bottom of the graph represent two typical classes. Only the second version of the HDoutliers algorithm (Figure 6 (b)- II) that utilises the clustering step was able to detect the top-centred micro cluster that contains three anomalous data instances. However, forming small clusters prior to the distance calculation is not always helpful in detecting micro clusters.

(3) Figure 6 (c)- II) shows a situation where even the second version of the HDoutliers algorithm fails in detecting micro clusters. The Leader algorithm in the HDoutliers algorithm uses a very small ball of a fixed radius to form clusters, and therefore, it now fails to capture the five points into a single cluster and instead generates three small clusters that are very close to one another. Both versions of the HDoutliers algorithm now fail to detect the micro cluster at the rightmost upper corner, because the dataset violates one of the major requirements of isolation of anomalous points or anomalous clusters. In stray, the value of k was set to 10. One can interpret the value of k as the maximum permissible size for a micro cluster. That is, for a small cluster to be a micro cluster, the number of data points in that cluster should be less than k. Otherwise, the cluster is considered a typical cluster.

(4) Figure 6 (d)- III) demonstrates the ability of detecting inliers. The HDoutliers algorithm also has this ability of detecting inliers only when there are isolated inliers that are free from anomalous neighbours. Both versions of the HDoutliers algorithm fail to detect the two inliers since they are very close to one another and thereby jointly project them as being anomalous.

(5) As explained in Section 3.2, Figure 6 (e)- II) shows how the clustering step of the second version of the HDoutliers algorithm can misguide the detection process and thereby increase the rate of false positives. The dense areas of the dataset are marked with density curves. Two typical clusters are visible, one at the leftmost upper corner and the other at the rightmost bottom corner. An inlier is also present in between the two typical classes. After forming cluster through the Leader algorithm, only one representative member is selected from each cluster for the nearest neighbour distance calculation. The selected member is now isolated and earns a very high anomalous score, leading the entire typical cluster at the rightmost bottom corner with 1,000 points to be identified as anomalous. In contrast, the stray algorithm is free from these problems because it does not involve any clustering step prior to the nearest neighbour distance calculation.

(6) As explained in Section 3.2, Figure 6 (f)- II) shows how the clustering step can increase the rate of false negatives. This dataset contains one typical class that is closely compacted in substance (the leftmost upper corner) and an obvious anomaly at the rightmost bottom corner. Since the typical class is a dense cluster, only a few data points are selected from the typical class for the nearest neighbour calculation. In this example, the clustering step substantially down-samples the original dataset, leading to a huge information loss in

the representation of the original dataset. The blue dots in Figure 6 (f)- II) represent the selected members from each cluster for nearest neighbour calculations. Now, the reduced sample size is not enough for a proper calculation of the anomalous threshold based on extreme value theory.

6 Usage

We applied our stray algorithm to a dataset obtained from an automated pedestrian counting system with 43 sensors in the city of Melbourne, Australia (City of Melbourne 2019; Wang 2018), to identify unusual pedestrian activities within the municipality. Identification of such unusual, critical behaviours of pedestrians at different city locations at different times of the day is important because it is a direct indication of a city's economic conditions, the related activities and the safety and convenience of the pedestrian experience (City of Melbourne 2019). It also guides and informs decision-making and planning. This case study also illustrates how the stray algorithm can be used to deal with other data structures, such as temporal data and streaming data using feature engineering.

6.1 Handling Temporal Data

For clear illustration, we limit our study period to one month from 1 December to 31 December 2018. Figure 7 shows the pedestrian counts at 43 locations in the city of Melbourne at different times of the day. Each scatterplot follows a negatively skewed distribution. In general, weekdays display a bimodal distribution, while weekends follow a unimodal distribution. Now, the aim is to detect days with unusual behaviours. Since this involves a large collection of scatterplots, manual monitoring is time-consuming and unusual behaviours are difficult to locate by visual inspection.

Detecting anomalous scatterplots from a large collection of scatterplots requires some pre-processing. In particular, to apply the stray algorithm, we need to convert this original dataset, with a large collection of scatterplots, into a high dimensional dataset. A simpler approach is to use features that describe the different shapes and patterns of the scatterplots. Computing features that describe meaningful shapes and patterns in a given scatterplot is straightforward with scagnostics (scatterplot diagnostics) developed by Wilkinson, Anand & Grossman (2005). For the current study, we select five features: outlying, convex, skinny, stringy and monotonic (Dang & Wilkinson 2014; Wilkinson, Anand & Grossman 2005). We specifically select these

features to address our use-case. Once we extract these five features from each scatterplot, we convert our original collection of scatterplots into a high-dimensional dataset with five dimensions and 31 data instances. Figure 8 provides feature-based representation of the original collection of scatterplots. Each point in this high-dimensional data space corresponds to a single scatterplot (or a day) in the original collection of scatterplots. In this high-dimensional space, the stray algorithm detects two anomalous points and they are marked in red colour in Figure 8. The corresponding scatterplots (or days) are marked in red colour in Figure 7. Visual inspection also confirms the anomalous behaviour of these two scatterplots. Both days, 1 December 2018 and 31 December 2018, display an unusual rise later in the day. One selected day, 31 December 2018, is an obvious anomaly since it is the New Year's Eve, and the associated fireworks at Southbank in the city of Melbourne attract many thousands of visitors. Further investigations regarding 1 December 2018, reveal that there was a musical concert at the Melbourne Cricket Ground from 8.00 pm and the unusual rise later in the day could be due to the concert participants.

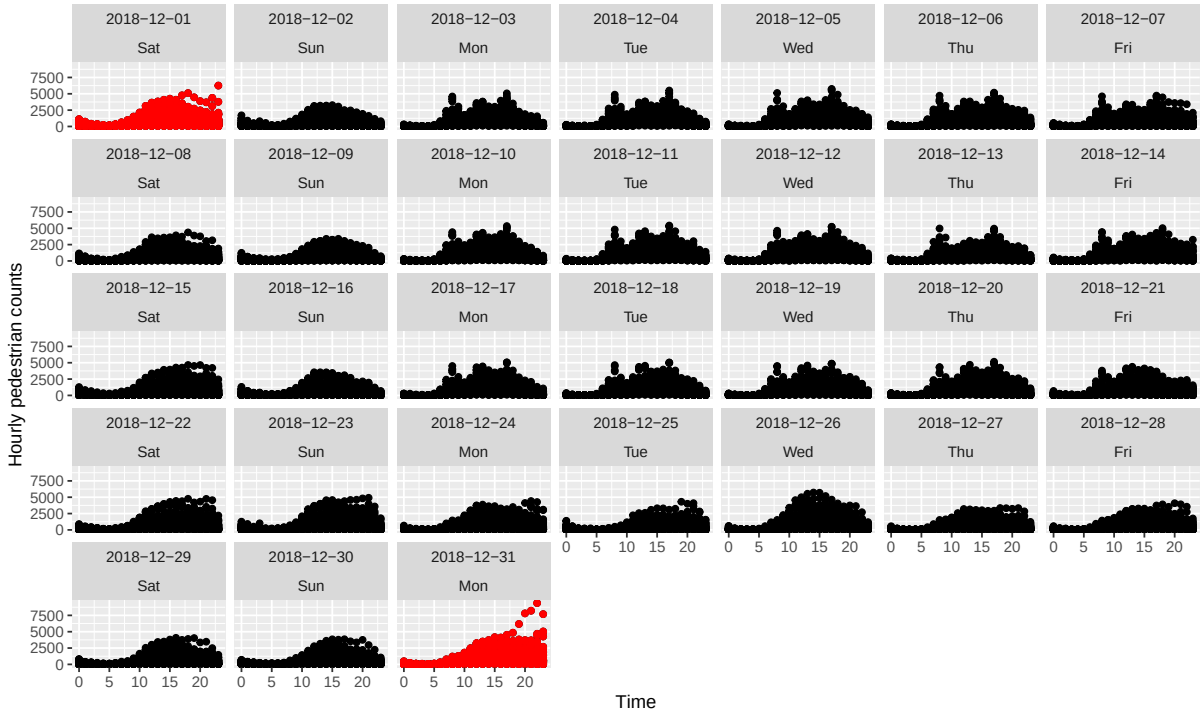

Figure 7: *Scatterplots of hourly pedestrian counts at 43 locations in the city Melbourne, Australia, from 1 December to 31 December 2018. Anomalous days detected by the stray algorithm using scagnostics are marked in red colour.*

After detecting the anomalous scatterplots or the days with anomalous pedestrian behaviours, further investigation is carried out for each day to detect the locations with anomalous behaviours within the selected day. Once we focus on one day, we obtain a collection of 43 time series with hourly pedestrian counts generated from the 43 sensors located at different geographical locations in the city (Figure 9). For this analysis, we extract seven time series features (similar

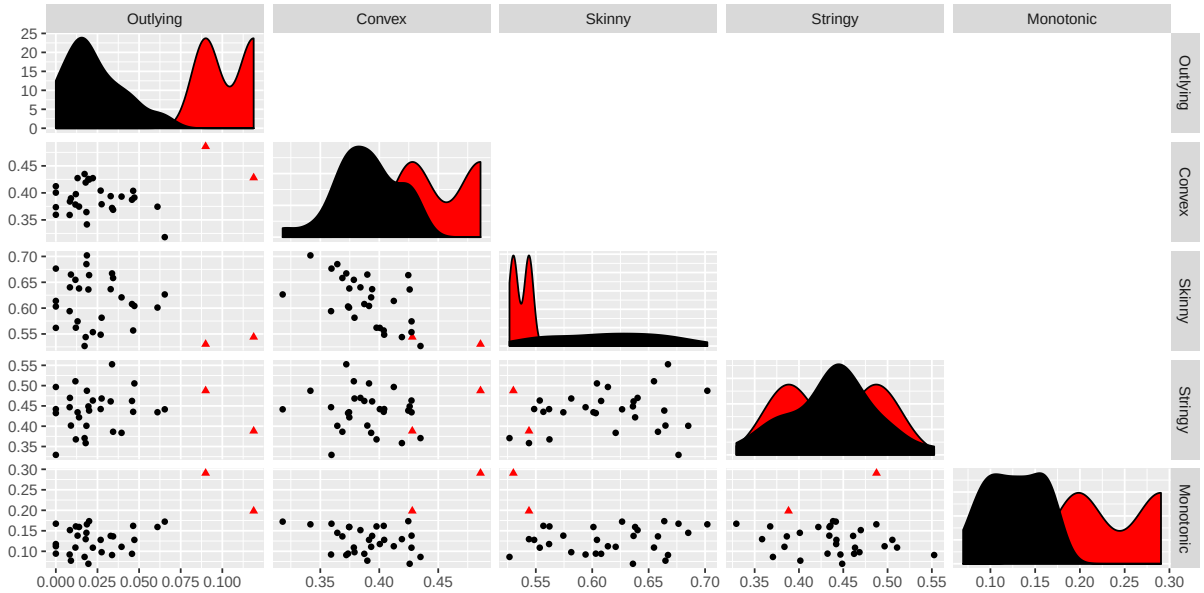

Figure 8: *Feature -based representation of the collection of scatterplots using Scagnostics. In each plot anomalies determined by the stray algorithm are represented by red colour.*

to Talagala et al. 2019a; Hyndman, Wang & Laptev 2015) and convert the original collection of time series into a high dimensional data space with seven dimensions and 43 data instances (Figure 10). Now, each point in this high-dimensional space correspond to a single time series (or sensor) in Figure 9. The stray algorithm declares one point as an anomalous point in this high dimensional data space. This point corresponds to the sensor positions at Southbank in Melbourne where New Year's Eve fireworks attract millions of spectators annually.

These types of findings play a critical role to make decisions about urban planning and management; to identify opportunities to improve city walkability and transport measures; to understand the impact of major events and other extreme conditions on pedestrian activity, and thereby assist in making decisions regarding security and resource requirements; and to plan and respond to emergency situations, etc.

6.2 Handling Streaming Data

Owing to the unsupervised nature of the stray algorithm, it can easily be extended for streaming data. A sliding window of fixed length can be used to deal with streaming data. Then, datasets in each window can be treated as a batch dataset (Talagala et al. 2019b) and the stray algorithm can be applied to each window to detect anomalies in the datasets defined by the corresponding window.

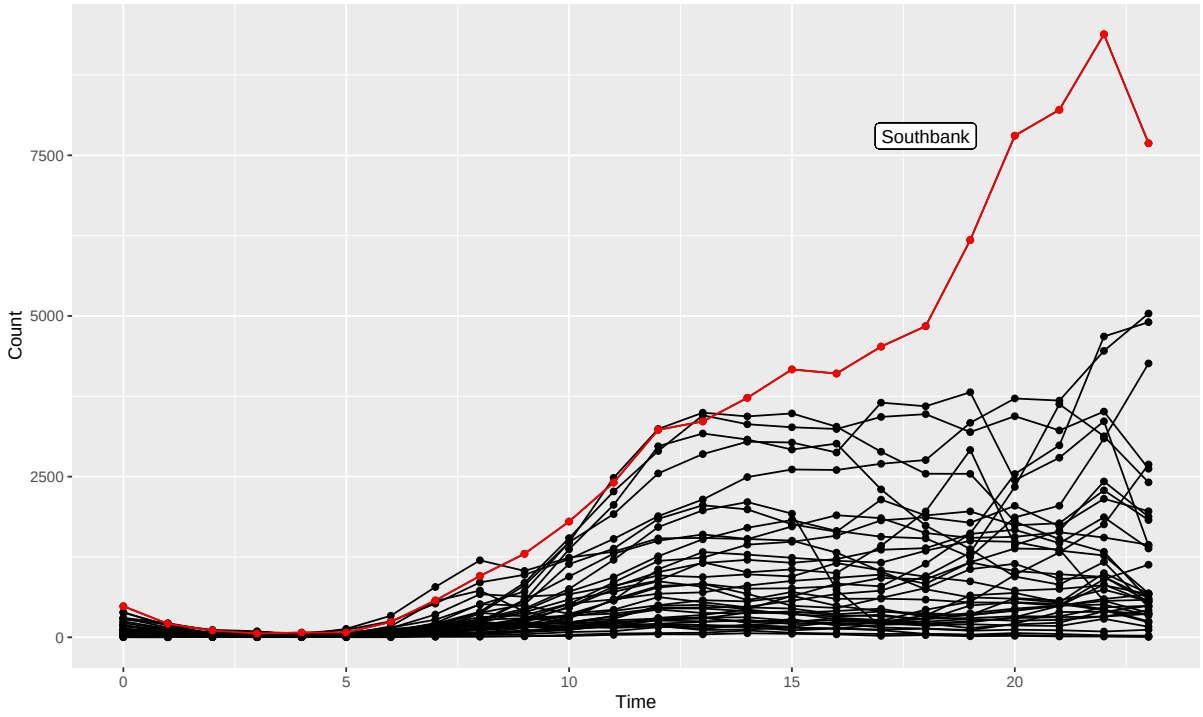

Figure 9: *Multivariate time series plot of hourly counts of pedestrians measured at 43 different sensors in the city of Melbourne, on 31 December 2018. The anomalous time series detected by the stray algorithm using time series features are marked in red colour.*

It also can be used to identify anomalous time series within a large collection of streaming temporal data. Let $W[t, t + w]$ represent a sliding window containing n number of individual time series of length w. First, we extract m features (similar to Hyndman, Wang & Laptev (2015) and Talagala et al. (2019b)) from each and every time series in this window. This step gives rise to an $n \times m$ feature matrix where each row now corresponds to a time series in the original collection of time series. Once we convert our original collection of time series into a high-dimensional dataset, we can apply the stray algorithm to identify anomalous points within this m-dimensional data space. The corresponding time series are then declared as anomalous series within the large collection of time series in the corresponding sliding window.

7 Conclusions and Further Research

The HDoutliers algorithm by Wilkinson (2017) is a powerful algorithm for detecting anomalies in high-dimensional data. However, it suffers from a few limitations that significantly hinder its ability to detect anomalies under certain situations. In this study, we propose an improved algorithm, the stray algorithm, that addresses these limitations. We define an anomaly here as an observation that deviates markedly from the majority with a large distance gap. We also

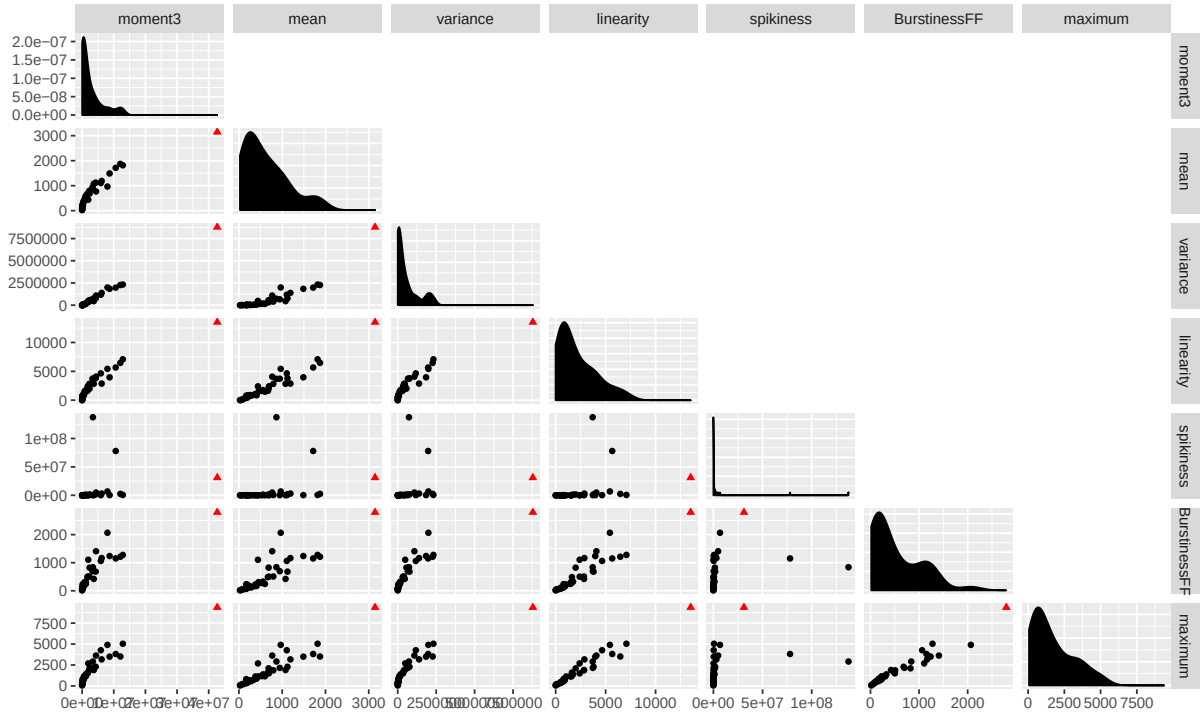

Figure 10: *Feature-based representation of the collection of time series on 31 December 2018. In each plot, anomalies determined by the stray algorithm are represented in red colour*

demonstrate how the stray algorithm can assist in detecting anomalies present in other data structures using feature engineering. In addition to a label, the stray algorithm also assigns an anomalous score to each data instance to indicate the degree of outlierness of each measurement.

While the HDoutliers algorithm is powerful, we have provided several classes of counterexamples in this paper where the structural properties of the data did not enable HDoutliers to detect certain types of outliers. We demonstrated on these counterexamples that the stray algorithm outperforms HDoutliers, in terms of both accuracy and computational time. It is certainly common practice to evaluate the strength of an algorithm using collections of test problems with various challenging properties. However, we acknowledge that these counterexamples are not diverse and challenging enough to enable us to comment about the unique strengths and weaknesses of these two algorithms, nor to generalise our findings to conclude that stray is always the superior algorithm. This study should be viewed as an attempt to simulate further investigation on the HDoutliers algorithm and its successors, with the ultimate goal to achieve further improvements across the entire problem space defined by various high-dimensional datasets. An important open research problem is therefore to assess the effectiveness of these algorithms across the the broadest possible problem space defined by different datasets with diverse properties (Kang, Hyndman & Smith-Miles 2017). It is an interesting question to explore the impact of other classes of problems with various structural properties affect the performance

of the stray algorithm and where its weaknesses might lie. This kind of instance space analysis (Smith-Miles et al. 2014) will enable further insights into improved algorithm design.

Anomaly detection problems commonly appear in many applications in different application domains. Therefore, it is hoped that different people with different knowledge levels will use the stray algorithm for many different purposes. Therefore, we expect future studies to develop interactive data visualisation tools that can enable exploring anomalies using a combination of graphical and numerical methods.

Supplementary Materials

R package `stray`: The `stray` package (Talagala, Hyndman & Smith-Miles 2019) consists of the implementation of the stray algorithm as described in this article. Version 0.1.0 of the package was used for the results presented in the article and is available from Github https://github.com/pridiltal/stray.

Data: To facilitate reproducibility and data reuse, the datasets used for this article are available in the open source R package `stray` (Talagala, Hyndman & Smith-Miles 2019)

Acknowledgements

This research was supported in part by the Monash eResearch Centre and eSolutions-Research Support Services through the use of the MonARCH (Monash Advanced Research Computing Hybrid) HPC Cluster.

References

Abuzaid, A, A Hussin & I Mohamed (2013). Detection of outliers in simple circular regression models using the mean circular error statistic. *Journal of Statistical Computation and Simulation* **83**(2), 269–277.

Aggarwal, CC (2017). *Outlier analysis*. Second edition. Cham, Switzerland : Springer.

Aggarwal, CC & PS Yu (2008). Outlier detection with uncertain data. In: *Proceedings of the 2008 SIAM International Conference on Data Mining*. SIAM, pp.483–493.

Ben-Gal, I (2005). "Outlier detection". In: *Data mining and knowledge discovery handbook*. Springer, pp.131–146.

Beygelzimer, A, S Kakadet, J Langford, S Arya, D Mount & S Li (2019). *FNN: Fast Nearest Neighbor Search Algorithms and Applications*. R package version 1.1.3. https://CRAN.R-project.org/package=FNN.

Breunig, MM, HP Kriegel, RT Ng & J Sander (2000). LOF: identifying density-based local outliers. In: *ACM Sigmod Record*. Vol. 29. 2. ACM, pp.93–104.

Burridge, P & AMR Taylor (2006). Additive outlier detection via extreme-value theory. *Journal of Time Series Analysis* **27**(5), 685–701.

Campos, GO, A Zimek, J Sander, RJ Campello, B Micenková, E Schubert, I Assent & ME Houle (2016). On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery* **30**(4), 891–927.

Cao, N, C Shi, S Lin, J Lu, YR Lin & CY Lin (2015). Targetvue: Visual analysis of anomalous user behaviors in online communication systems. *IEEE Transactions on Visualization and Computer Graphics* **22**(1), 280–289.

Chandola, V, A Banerjee & V Kumar (July 2009). Anomaly detection: A survey. *ACM Computing Surveys* **41**(3), 1–58.

City of Melbourne (2019). *Pedestrian Volume in Melbourne*. Last accessed 2019-07-23. http://www.pedestrian.melbourne.vic.gov.au.

Clifton, DA, S Hugueny & L Tarassenko (2011). Novelty detection with multivariate extreme value statistics. *Journal of Signal Processing Systems* **65**(3), 371–389.

Dang, TN & L Wilkinson (2014). Transforming scagnostics to reveal hidden features. *IEEE Transactions on Visualization and Computer Graphics* **20**(12), 1624–1632.

Elseberg, J, S Magnenat, R Siegwart & A Nüchter (2012a). Comparison of nearest-neighbor-search strategies and implementations for efficient shape registration. *Journal of Software Engineering for Robotics (JOSER)* **3**(1), 2–12.

Elseberg, J, S Magnenat, R Siegwart & A Nüchter (2012b). Comparison of nearest-neighbor-search strategies and implementations for efficient shape registration. *Journal of Software Engineering for Robotics* **3**(1), 2–12.

Embrechts, P, C Klüppelberg & T Mikosch (2013). *Modelling Extremal Events: for Insurance and Finance*. Stochastic Modelling and Applied Probability. Springer Berlin Heidelberg.

Fraley, C (2018). *HDoutliers: Leland Wilkinson's Algorithm for Detecting Multidimensional Outliers*. R package version 1.0. https://CRAN.R-project.org/package=HDoutliers.

Galambos, J, J Lechner & E Simiu (2013). *Extreme Value Theory and Applications: Proceedings of the Conference on Extreme Value Theory and Applications, Volume 1 Gaithersburg Maryland 1993*.

Extreme Value Theory and Applications: Proceedings of the Conference on Extreme Value Theory and Applications, Gaithersburg, Maryland, 1993. Springer US.

Gao, J, W Hu, ZM Zhang, X Zhang & O Wu (2011). RKOF: robust kernel-based local outlier detection. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, pp.270–283.

Goldstein, M & S Uchida (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PlOS ONE* **11**(4), e0152173.

Grubbs, FE (1969). Procedures for detecting outlying observations in samples. *Technometrics* **11**(1), 1–21.

Gupta, M, J Gao, CC Aggarwal & J Han (2014). Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and Data Engineering* **26**(9), 2250–2267.

Hartigan, JA & J Hartigan (1975). *Clustering Algorithms*. Vol. 209. Wiley New York.

Hodge, V & J Austin (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review* **22**(2), 85–126.

Hofmann, H, H Wickham & K Kafadar (2017). Value plots: Boxplots for large data. *Journal of Computational and Graphical Statistics* **26**(3), 469–477.

Hyndman, RJ (1996). Computing and graphing highest density regions. *The American Statistician* **50**(2), 120–126.

Hyndman, RJ, E Wang & N Laptev (2015). Large-scale unusual time series detection. In: *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pp.1616–1619.

Jin, W, AK Tung, J Han & W Wang (2006). Ranking outliers using symmetric neighborhood relationship. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, pp.577–593.

Jouan-Rimbaud, D, E Bouveresse, D Massart & O De Noord (1999). Detection of prediction outliers and inliers in multivariate calibration. *Analytica Chimica Acta* **388**(3), 283–301.

Kang, Y, RJ Hyndman & K Smith-Miles (2017). Visualising forecasting algorithm performance using time series instance spaces. *International Journal of Forecasting* **33**(2), 345–358.

Kanungo, T, DM Mount, NS Netanyahu, CD Piatko, R Silverman & AY Wu (2002). An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis & Machine Intelligence* (7), 881–892.

Leigh, C, O Alsibai, RJ Hyndman, S Kandanaarachchi, OC King, JM McGree, C Neelamraju, J Strauss, PD Talagala, RD Turner, K Mengersen & EE Peterson (2019). A framework for automated anomaly detection in high frequency water-quality data from in situ sensors. *Science of the Total Environment* **664**, 885–898.

Liu, S, D Maljovec, B Wang, PT Bremer & V Pascucci (2016). Visualizing high-dimensional data: Advances in the past decade. *IEEE Transactions on Visualization and Computer Graphics* **23**(3), 1249–1268.

Madsen, JH (2018). *DDoutlier: Distance and Density-Based Outlier Detection*. R package version 0.1.0. https://CRAN.R-project.org/package=DDoutlier.

Novotny, M & H Hauser (2006). Outlier-preserving focus+ context visualization in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics* **12**(5), 893–900.

Pimentel, MA, DA Clifton, L Clifton & L Tarassenko (2014). A review of novelty detection. *Signal Processing* **99**, 215–249.

R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. https://www.R-project.org/.

Sabahi, F & A Movaghar (2008). Intrusion detection: A survey. In: *3rd International Conference on Systems and Networks Communications-ICSNC'08*. IEEE, pp.23–26.

Schwarz, KT (2008). *Wind dispersion of carbon dioxide leaking from underground sequestration, and outlier detection in eddy covariance data using extreme value theory*. ProQuest.

Shahid, N, IH Naqvi & SB Qaisar (2015). Characteristics and classification of outlier detection techniques for wireless sensor networks in harsh environments: a survey. *Artificial Intelligence Review* **43**(2), 193–228.

Smith-Miles, K, D Baatar, B Wreford & R Lewis (2014). Towards objective measures of algorithm performance across instance space. *Computers & Operations Research* **45**, 12–24.

Sundaram, S, IGD Strachan, DA Clifton, L Tarassenko & S King (2009). Aircraft engine health monitoring using density modelling and extreme value statistics. In: *Proceedings of the 6th International Conference on Condition Monitoring and Machine Failure Prevention Technologies*.

Talagala, PD, RJ Hyndman, K Smith-Miles, S Kandanaarachchi & MA Muñoz (2019a). Anomaly Detection in Streaming Nonstationary Temporal Data. *Journal of Computational and Graphical Statistics*, 1–21.

Talagala, PD, RJ Hyndman, C Leigh, K Mengersen & K Smith-Miles (2019b). A feature-based framework for detecting technical outliers in water-quality data from in situ sensors. *arXiv preprint arXiv:1902.06351*.

Talagala, PD, RJ Hyndman & K Smith-Miles (2019). *stray: Anomaly Detection in High Dimensional and Temporal Data*. R package version 0.1.0.9000.

Tang, J, Z Chen, AWC Fu & DW Cheung (2002). Enhancing effectiveness of outlier detections for low density patterns. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, pp.535–548.

Unwin, A (2019). Multivariate outliers and the O3 Plot. *Journal of Computational and Graphical Statistics*, 1–11.

Wang, E (2018). *rwalkr: API to Melbourne Pedestrian Data*. R package version 0.4.0. https://CRAN.R-project.org/package=rwalkr.

Weissman, I (1978). Estimation of parameters and large quantiles based on the k largest observations. *Journal of the American Statistical Association* **73**(364), 812–815.

Wilkinson, L (2017). Visualizing big data outliers through distributed aggregation. *IEEE Transactions on Visualization and Computer Graphics* **24**(1), 256–266.

Wilkinson, L, A Anand & R Grossman (2005). Graph-theoretic scagnostics. In: *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* IEEE, pp.157–164.

Williams, KT (2016). "Local parametric density-based outlier detection and ensemble learning with applications to malware detection". PhD thesis. The University of Texas at San Antonio.

Zhang, R (2017). *Performance of kd-tree vs brute-force nearest neighbor search on GPU?* Computational Science Stack Exchange. URL:https://scicomp.stackexchange.com/q/26873 (version: 2017-05-13).

Zhang, W, J Wu & J Yu (2010). An improved method of outlier detection based on frequent pattern. In: *Information Engineering (ICIE), 2010 WASE International Conference on*. Vol. 2. IEEE, pp.3–6.

Chapter 3

Anomaly Detection in Streaming Non-stationary Temporal Data

This article has been accepted for publication in the *Journal of Computational and Graphical Statistics* and is currently in press

Full paper is available at - https://www.tandfonline.com/doi/abs/10.1080/10618600.2019.1617160?journalCode=ucgs20

Chapter 4

A Feature-Based Procedure for Detecting Technical Outliers in Water-Quality Data from *in situ* Sensors

This article has been revised and resubmitted to the "Water Resources Research* for possible publication. The work is based on the collaborative research project carried out with the Queensland University of Technology and the Queensland Department of Environment and Science, Great Barrier Reef Catchment Loads Monitoring Program from April to July 2018.

A feature-based procedure for detecting technical outliers in water-quality data from in situ sensors

Priyanga Dilini Talagala[1,2], Rob J. Hyndman[1,2], Catherine Leigh[1,3], Kerrie Mengersen[1,4], Kate Smith-Miles[1,5]

[1]ARC Centre of Excellence for Mathematics and Statistical Frontiers (ACEMS), Australia
[2]Department of Econometrics and Business Statistics, Monash University, Australia
[3]Institute for Future Environments, Science and Engineering Faculty, Queensland University of Technology, Australia
[4]School of Mathematical Sciences, Science and Engineering Faculty, Queensland University of Technology, Australia
[5]School of Mathematics and Statistics, University of Melbourne, Australia

Key Points:

- Our feature-based procedure starts by applying different statistical transformations to water-quality data to highlight outliers in high dimensional space
- Density and distance-based unsupervised outlier scoring techniques were applied to detect outliers due to technical issues with the sensors
- An approach based on extreme value theory was then used to calculate outlier thresholds

Corresponding author: Priyanga Dilini Talagala, `dilini.talagala@monash.edu`

Abstract
Outliers due to technical errors in water quality data from *in situ* sensors can reduce data quality and have a direct impact on inference drawn from subsequent data analysis. However, outlier detection through manual monitoring is infeasible given the volume and velocity of data the sensors produce. Here, we introduce an automated procedure, named oddwater that provides early detection of outliers in water-quality data from *in situ* sensors caused by technical issues. Our oddwater procedure is used to first identify the data features that differentiate outlying instances from typical behaviours. Then statistical transformations are applied to make the outlying instances stand out in a transformed data space. Unsupervised outlier scoring techniques are applied to the transformed data space and an approach based on extreme value theory is used to calculate a threshold for each potential outlier. Using two datasets obtained from *in situ* sensors in rivers flowing into the Great Barrier Reef lagoon, Australia, we show that oddwater successfully identifies outliers involving abrupt changes in turbidity, conductivity and river level, including sudden spikes, sudden isolated drops and level shifts, while maintaining very low false detection rates. We have implemented this oddwater procedure in the open source R package `oddwater`.

1 Introduction

Water-quality monitoring traditionally relies on water samples collected manually. The samples are then analyzed within laboratories to determine the water-quality variables of interest. This type of rigorous laboratory analysis of field-collected samples is crucial in making natural resources management decisions that affect human welfare and environmental conditions. However, with the rapid advances in hardware technology, the use of *in situ* water-quality sensors positioned at different geographic sites is becoming an increasingly common practice used to acquire real-time measurements of environmental and water-quality variables. Though only a subset of the required water-quality variables can be measured by these sensors, they have several advantages. Their ability to collect large quantities of data and to archive historic records allows for deeper analysis of water-quality variables to improve understanding about field conditions and water-quality processes (Glasgow et al., 2004). Near-real-time monitoring also allows operators to identify and respond to potential issues quickly and thus manage the operations efficiently. Further, the use of *in situ* sensors can greatly reduce the labor involved in field sampling and laboratory analysis.

Water-quality sensors are exposed to changing environments and extreme weather conditions, and thus are prone to errors, including failure. Automated detection of outliers in water-quality data from *in situ* sensors has therefore captured the attention of many researchers both in the ecology and data science communities (Hill et al., 2009; Archer et al., 2003; Raciti et al., 2012; McKenna et al., 2007; Koch & McKenna, 2010). This problem of outlier detection in water-quality data from *in situ* sensors can be divided into two sub-topics according to their focus: (1) identifying errors in the data due to issues unrelated to water events per se, such as technical aberrations, that make the data unreliable and untrustworthy; and (2) identifying real events (e.g. rare but sudden spikes in turbidity associated with rare but sudden high-flow events). Both problems are equally important when making natural resource management decisions that affect human welfare and environmental conditions. Problem 1 can also be considered as a data preprocessing phase before addressing Problem 2.

In this work we focus on Problem 1, i.e. detecting unusual measurements caused by technical errors that make data unreliable and untrustworthy, and affect performance of any subsequent data analysis under Problem 2. According to Yu (2012), the degree of confidence in the sensor data is one of the main requirements for a properly defined

environmental analysis procedure. For instance, researchers and policy makers are unable to use water-quality data containing technical outliers with confidence for decision making and reporting purposes, because erroneous conclusions regarding the quality of the water being monitored could ensue, leading, for example, to inappropriate or unnecessary water treatment, land management or warning alerts to the public (Kotamäki et al., 2009; Rangeti et al., 2015). Missing values and corrupted data can also have an adverse impact on water-quality model building and calibration processes (Archer et al., 2003). Early detection of these technical outliers will limit the use of corrupted data for subsequent analysis. For instance, it will limit the use of corrupted data in real-time forecasting and online applications such as on-line drinking water-quality monitoring and early warning systems (Storey et al., 2011), predicting algal bloom outbreaks leading to fish kill events and potential human health impacts, forecasting water level and currents etc. (Glasgow et al., 2004; Archer et al., 2003; Hill & Minsker, 2006). However, because data arrive near continuously at high speed in large quantities, manual monitoring is highly unlikely to be able to capture all the errors. These issues have therefore increased the importance of developing automated methods for early detection of outliers in water-quality data from *in situ* sensors (Hill et al., 2009).

Different statistical approaches are available to detect outliers in water-quality data from *in situ* sensors. For example, Hill and Minsker (2006) addressed the problem of outlier detection in environmental sensors using regression-based time series models. In this work they addressed the scenario as a univariate problem. Their prediction models are based on four data-driven methods: naive, clustering, perceptron, and Artificial Neural Networks (ANN). Measurements that fell outside the bounds of an established prediction interval were declared as outliers. They also considered two strategies: anomaly detection (AD) and anomaly detection and mitigation (ADAM) for the detection process. ADAM replaces detected outliers with the predicted value prior to the next predictions whereas AD simply uses the previous measurements without making any alteration to the detected outliers. These types of data-driven methods develop models using sets of training examples containing a feature set and a target output. Later, Hill et al. (2009) addressed the problem by developing three automated anomaly detection methods using dynamic Bayesian networks (DBN) and showed that DBN-based detectors using either robust Kalman filtering or Rao-Blackwellized particle filtering, outperformed that of Kalman filtering.

Another common approach for detecting outliers in environmental sensor data is based on residuals (the differences between predicted and actual values). Due to the ability of ANNs to model a wide range of complex non-linear phenomena, Moatar, Fessant, and Poirel (1999) used ANN techniques to detect anomalies such as abnormal values, discontinuities, and drifts in pH readings. After developing the pH model, the Student t-test and the cumulative Page–Hinkley test were applied to detect changes in the mean of the residuals to detect measurement error occurring over short periods of time. The work was later expanded to a multivariate scenario with some additional water-quality variables including dissolved oxygen, electrical conductivity, pH and temperature (Moatar et al., 2001). Their proposed algorithm used both deterministic and stochastic approaches for the model building process. Observed data were then compared with the model forecasts using a set of classical statistical tests to detect outliers, demonstrating the effectiveness and advantages of the multimodel approach. Later, Archer et al. (2003) proposed a method to detect failures in the water-quality sensors due to biofouling based on a sequential likelihood ratio test. Their method also had the ability to provide estimates of biofouling onset time, which was useful for the subsequent step of outlier correction.

A common feature of all of the above methods is that they are usually employed in a supervised or semi-supervised context and thus require training data pre-labeled with known outliers or data that are free from the anomalous features of interest. In many cases, however, not all the possible outliers are known in advance and can arise spon-

taneously as new outlying behaviors during the test phase. In such situations, supervised methods may fail to detect those outliers. Semi-supervised methods are also unsuitable for certain applications due to the unavailability of training data containing only typical instances that are free from outliers (Goldstein & Uchida, 2016). The datasets that we consider in this paper suffer from both of these limitations highlighting the need for a more general approach.

This paper develops a method for detecting technical outliers in water-quality data derived from *in situ* sensors. Prior work by Leigh et al. (2019) emphasises the importance of different anomaly types and end-user needs and provides the starting point for constructing a framework for automated anomaly detection in high frequency water-quality data from *in situ* sensors. Their work briefly introduced unsupervised feature based methods for detecting technical-outliers in such data. The present paper differs substantially from Leigh et al. (2019) as (1) the unsupervised feature based procedure we present for detecting technical-outliers in high frequency water-quality data measured by *in situ* sensors is its sole focus (2) the unsupervised feature based procedure is fully elaborated in both details and depth and (3) the experimental results are enhanced through emphasis on the multivariate capabilities of the unsupervised feature based procedure. Furthermore, we focus on outliers involving abrupt changes in value, including sudden spikes, sudden isolated drops and level shifts (high priority outliers as described in Leigh et al. (2019)) rather than the broader suite considered by Leigh et al. (2019).

First, we present in detail our unsupervised feature based procedure that provides early detection of technical outliers in water-quality data from *in situ* sensors. Rule-based methods are also incorporated into the procedure to flag occurrences of impossible, out-of-range, and missing values. Second, we provide a comparative analysis of the efficacy and reliability of both density- and nearest neighbor distance-based outlier scoring techniques. Third, we introduce an R (R Core Team, 2018) package, `oddwater` (Talagala & Hyndman, 2018) that implements the feature-based procedure and related functions. Further, to facilitate reproducibility and reuse of the results presented in this paper we have made all of the code and associated datasets available on zenodo[1].

Our feature-based procedure has many advantages: (1) it can take the correlation structure of the water-quality variables into account when detecting outliers; (2) it can be applied to both univariate and multivariate problems; (3) the outlier scoring techniques that we consider are unsupervised, data-driven approaches and therefore do not require training datasets for the model building process, and can be extended easily to other time series from other sites; (4) the outlier thresholds have a probabilistic interpretation as they are based on extreme value theory; (5) the approach has the ability to deal with irregular (unevenly spaced) time series; and (6) it can easily be extended to streaming data. In contrast to a batch scenario, which assumes that the entire dataset is available prior to the analysis with the focus on detecting complete events, the streaming data scenario gives many additional challenges due to high velocity, unbounded, nonstationary data with incomplete events (Hill et al., 2009; Talagala et al., 2018). In this paper, although our oddwater procedure is introduced as a batch method it can easily be extended to streaming data such that it can provide near-real-time support using a sliding window technique.

2 Materials and Methods

Our unsupervised feature-based procedure for detecting outliers in water-quality data from *in situ* sensors has six main steps (Figure 1), and the structure of this section is organised accordingly. For easy reference, we named our unsupervised feature-based

[1] DOI: 10.5281/zenodo.2890469

procedure as oddwater procedure, which stands for **O**utlier **D**etection in **D**ata from **WA-TER**-quality sensors

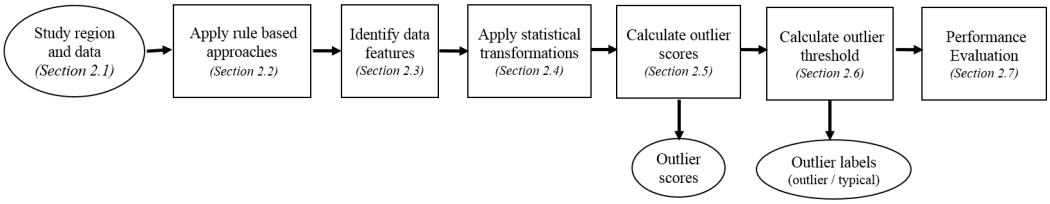

Figure 1. Unsupervised feature-based procedure, named oddwater procedure for outlier detection in water quality data from *in situ* sensors. Squares represents the main steps involved. Circles correspond to input and output.

2.1 Study region and data

To evaluate the effectiveness of our oddwater procedure we considered a challenging real-world problem of monitoring water-quality using *in situ* sensors in a natural river system. This is challenging because the system is susceptible to a wide range of environmental, biological and human impacts that can lead to variation in water-quality and affect the technological performance of the sensors. For comparison, we evaluated two study sites, Sandy Creek and Pioneer River (PR), both in the Mackay-Whitsunday region of northeastern Australia (Mitchell et al., 2005). These two rivers flow into the Great Barrier Reef lagoon, and have catchment areas of 1466 km^2 and 326 km^2, respectively. In this region, the wet season typically occurs from December to April and is dominated by higher rainfall and air temperatures, whereas the dry season typically occurs from May to November with lower rainfall and air temperatures (McInnes et al., 2015). The sensors at these two sites are housed within a monitoring station on the river banks. Water is pumped from the rivers to the stations approximately every 60 or 90 minutes to take measurements of various water-quality variables that are logged by the sensors. Here we focused on three water-quality variables: turbidity(NTU), conductivity (strictly, specific conductance at 25^0C; μS/cm) and river level (m).

The water-quality data obtained from *in situ* sensors located at Sandy Creek were available from 12 March 2017 to 12 March 2018. The data set included 5402 recorded points. These time series were irregular (i.e. the frequency of observations was not constant) with a minimum time gap of 10 minutes and a maximum time gap of around 4 hours. The data obtained from Pioneer River were available from 12 March 2017 to 12 March 2018, and included 6280 recorded points. Many missing values were observed during the initial part of all three series, turbidity, conductivity and river level, at Pioneer River. With the help of a group of water-quality experts, familiar with the study region and with over 40 years of combined knowledge of river water quality, observations were labeled as outliers or not, with the aim of evaluating the performance of the procedure. Our Shiny web application available through the *oddwater* R package was used during the labeling process to pinpoint observations and provide greater visual insight into the data. Using this interactive visualization tool and expert knowledge, the ground-truth labels were decided by consensus vote.

2.2 Apply rule-based approaches

Following Thottan and Ji (2003), we incorporated simple rules into our oddwater procedure to detect outliers such as out-of-range values, impossible values (e.g. negative values) and missing values, and labeled them prior to applying the statistical transformations introduced in Section 2.4.

If a sensor reading was outside the corresponding sensor detection range it was marked as an outlier. Negative readings are also inaccurate and impossible for river turbidity, conductivity and level. We therefore imposed a simple constraint on the algorithm to filter these values and mark them as outliers. Missing values are also frequently encountered in water-quality sensor data (Rangeti et al., 2015). We detected missing values by calculating the time gaps between readings. If a gap exceeded the maximum allowable time difference between any two consecutive readings, the corresponding time stamp was then marked as an outlier due to missingness. Here, the maximum allowable time difference was set at 180 minutes, given that the water-quality measurements were set to be taken at most every 90 minutes (measurements were often taken at higher frequencies during high-flow events, e.g. every 10-15 minutes, and occasionally as one-off measurements at times of interest to water managers).

2.3 Identify data features

After labeling out-of-range, impossible and missing values as outliers, further investigation was done with the remaining observations. We initiated this investigation by identifying common characteristics or patterns of the possible types of outliers in water-quality data that would differentiate them from typical instances or events. For turbidity, for example, "extreme" deviations upward are more likely than deviations downwards (Panguluri et al., 2009). The opposite is true for conductivity (Tutmez et al., 2006). Further, in a turbidity time series a sudden isolated upward shift (spike) is a point outlier (a single observation that is surprisingly large, independent of the neighboring observations (Goldstein & Uchida, 2016)), but if the sudden upward shift is followed by a gradually decaying tail then it becomes part of the typical behavior. For river level, rates of rise are often fast compared with fall rates. In general, isolated data points that are outside the general trend are outliers. Further, natural water processes under typical conditions generally tend to be comparatively slow; sudden changes therefore mostly correspond to outlying behaviors. Hereafter, these characteristics will be referred to as 'data features'.

2.4 Apply statistical transformations

After identifying the data features, different statistical transformations were applied to the time series to highlight different types of outliers, focusing on sudden isolated spikes, sudden isolated drops, sudden shifts, and clusters of spikes (Table 1) that deviate from the typical characteristics of each variable (Leigh et al., 2019).

In this work we considered the outlier detection problem in a multivariate setting. By applying different transformations on water-quality variables we converted our original problem of outlier detection in the temporal context to a non-temporal context through a high dimensional data space with three dimensions defined by the three variables: turbidity, conductivity and river level. Different transformations were applied on different axes of the three dimensional data space resulting in different data patterns. We evaluated the performance of the transformations (Dang & Wilkinson, 2014) using the maximum separability of the two classes: outliers and typical points in the three dimensional data space. For example, in the data obtained from Sandy Creek, the one sided derivative transformation clearly separated all of the target outlying points from the typical points, shown as either red triangles (corresponding to outliers) or green squares (corresponding to the immediate neighbours of outliers) (Figure 2(e)). To provide a better visual illustration, in Figure 2, we present only the two dimensional data space defined

Table 1. Transformation methods used to highlight different types of outliers in water-quality sensor data. Let Y_t represent an original series from one of the three variables: turbidity, conductivity and level at time t.

Transformation	**Formula**	**Data Feature**	**Focus**
Log transformation	$\log(y_t)$	High variability of the data.	To stabilize the variance across time series and make the patterns more visible (e.g. level shifts)
First difference	$\log(y_t/y_{t-1})$	Isolated spikes (in both positive and negative directions) that are outside the general trend are considered as outliers. Under typical behavior, sudden upward (downward) shifts are possible for turbidity (conductivity), but their rate of fall (rise) is generally slower than the rate of rise (fall).	To separate isolated spikes from the general upward/downward trend patterns.
Time gap	Δt		To identify missing values.
First derivative	$x_t = \log(y_t/y_{t-1})/\Delta t$	Data are unevenly spaced time series.	To handle irregular time series. Data points with large gaps will get small value. Large gaps indicate the lack of information to make a claim regarding the points.
One sided derivative			
Turbidity or level	$\min\{x_t, 0\}$	Extreme upward trend in turbidity and level under typical behavior.	To separate spikes from typical upward trends.
Conductivity	$\max\{x_t, 0\}$	Extreme downward trend in conductivity under typical behavior.	To separate isolated drops from typical downward trends.
Rate of change	$(y_t - y_{t-1})/y_t$	High or low variability in the data.	To detect change points in variance.
Relative difference	$y_t - (1/2)(y_{t-1} + y_{t+1})$	Natural processes are comparatively slow. Sudden changes (upward or downward movements) typically correspond to outlying instances.	To detect sudden changes (both upward and downward movements)

by turbidity and conductivity; however our actual data space is three dimensional. In this work our focus was to evaluate whether each point in time is an outlier or not, such that an alarm could be triggered in the presence of an outlier. However, it was not our interest to investigate which variable(s) is (are) responsible for the outlier in time. Therefore, in Figure 2, a point is marked as an outlier in the two dimensional space if at least one variable corresponding to that point was labelled as an outlier by the water-quality experts.

When the transformation involves both the current value Y_t and the lagged value Y_{t-1} (as in the first difference, first derivative, and one sided derivatives), the neighboring points can emerge as outliers instead of the actual outlying point. For an example, if an outlier occurs at time point t, then the two values derived from the first derivative transformation $((y_t - y_{t-1})$ and $(y_{t+1} - y_t))$ get highlighted as outlying values because they both involve y_t. That is each outlying instance is now represented by two consecutive values under the first derivative transformation. The goal of the one sided derivative transformation is to filter one high value for each outlying instance. However the high values obtained could correspond to either the actual outlying time point or the neighboring time point, because each transformed value is derived from two consecutive observations. If the primary focus of detecting technical outliers is to alert managers of sensor failures, then it will be inconsequential if the alarm is triggered either at the actual time point corresponding to the outlier or at the next immediate time point. However if the purpose is different, such as producing a trustworthy dataset by labeling or correcting detected outliers, then additional conditions should be imposed to ensure that the time points declared as outliers correspond to the actual outlying points and not to their immediate neighboring points.

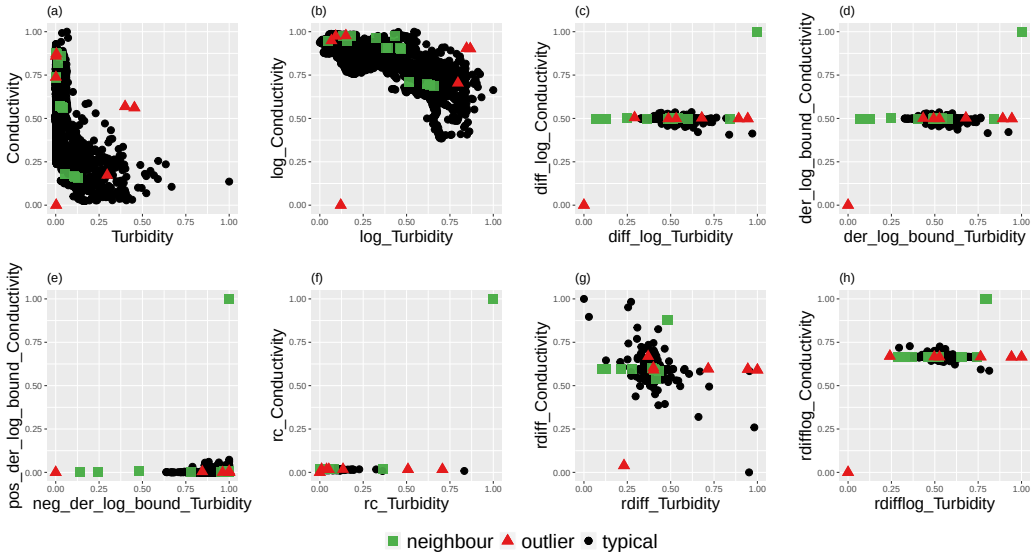

Figure 2. Bivariate relationships between transformed series of turbidity and conductivity measured by *in situ* sensors at Sandy Creek. In each scatter plot, outliers determined by water-quality experts are shown in red, while typical points are shown in black. Neighboring points are marked in green. (a) Original series, (b) Log transformation, (c) First difference, (d) First derivative, (e) One sided derivative, and (f) Rate of change, (g) Relative difference (for original series), (h) Relative difference (for log transformed series). In each scatter plot, data are normalised such that they are bounded by the unit hypercube.

2.5 Calculate outlier scores

We considered eight unsupervised outlier scoring techniques for high dimensional data, involving nearest neighbor distances or densities of the observations and applied them to the three dimensional data space defined by the three variables: turbidity, conductivity and river level. Methods based on k-nearest neighbor distances (where $k \in Z^+$) were the NN-HD algorithm (details of this algorithm, which was inspired by HDoutliers algorithm (Wilkinson, 2018) are provided in Supporting Information), KNN-AGG and KNN-SUM algorithms (Angiulli & Pizzuti, 2002; Madsen, 2018) and Local Distance-based Outlier Factor (LDOF) algorithm (Zhang et al., 2009), which calculate the outlier score under the assumption that any outlying point (or outlying clusters of points) in the data space is (are) isolated; therefore the outliers are those points having the largest k-nearest neighbor distances. In contrast, the density based Local Outlier Factor (LOF) (Breunig et al., 2000), Connectivity-based Outlier Factor (COF) (Tang et al., 2002), Influenced Outlierness (INFLO) (Jin et al., 2006) and Robust Kernel-based Outlier Factor (RKOF) (Gao et al., 2011) algorithms calculate an outlier score based on how isolated a point is with respect to its surrounding neighbors, and therefore the outliers are those points having the lowest densities (see Supporting Information for detail). Each algorithm assigns outlier scores for all of the data points in the high dimensional space that described the degree of outlierness of the individual data points such that outliers are those points having the largest scores (Kriegel et al., 2010; Shahid et al., 2015). This step allowed us to set a data driven threshold (Section 2.6) for the outlier scores, to select the most relevant outliers (Chandola et al., 2009).

2.6 Calculate outlier threshold

Following Schwarz (2008), Burridge and Taylor (2006) and Wilkinson (2018), we used extreme value theory (EVT) to calculate a separate outlier threshold for each set of outlier scores calculated using a given unsupervised outlier scoring technique (introduced in Section 2.5) and assign a bivariate label for each point either as an outlier or typical point. Thus, 8 outlier scoring techniques resulted 8 different thresholds for a given dataset. The threshold calculation process started from a subset of data containing 50% of observations with the smallest outlier scores, under the assumption that this subset contained the outlier scores corresponding to typical data points and the remaining subset contained the scores corresponding to the possible candidates for outliers. Following Weissman's spacing theorem (Weissman, 1978), the algorithm then fit an exponential distribution to the upper tail of the outlier scores of the first subset, and computed the upper $1-\alpha$ (in this work α was set to 0.05) points of the fitted cumulative distribution function, thereby defining an outlying threshold for the next outlier score. From the remaining subset the algorithm then selected the point with the smallest outlier score. If this outlier score exceeded the cutoff point, all the points in the remaining subset were flagged as outliers and searching for outliers ceased. Otherwise, the point was declared as a non-outlier and was added to the subset of the typical points. The threshold was then updated by including the latest addition. The searching algorithm continued until an outlier score was found that exceeded the latest threshold (Schwarz, 2008). We performed this threshold calculation under the assumption that the distribution of outlier scores produced by each of the eight unsupervised outlier scoring techniques for high dimensional data was in the maximum domain of attraction of the Gumbel distribution which consists of distribution functions with exponentially decaying tails including the exponential, gamma, normal and log-normal (Embrechts et al., 2013).

2.7 Performance evaluation

In this paper, we focused on high priority outliers, as described in Leigh et al. (2019), in which importance ranking of different outlier types was done by taking into account the end-user goals and the potential impact of outliers going undetected. However, it is

beyond the scope of this paper to discuss in detail the different types of outliers and their importance ranking. For more detail, we refer the reader to Leigh et al. (2019). We performed an experimental evaluation on the accuracy and computational efficiency of our oddwater procedure with respect to the eight outlier scoring techniques, using the different transformations (Table 1) and different combinations of variables (turbidity, conductivity and river level). These experimental combinations were evaluated with respect to common measures for binary classification based on the values of the confusion matrix which summarizes the false positives (FP; i.e. when a typical observation is misclassified as an outlier), false negatives (FN; i.e. when an actual outlier is misclassified as a typical observation), true positives (TP; i.e. when an actual outlier is correctly classified), and true negatives (TN; i.e. when an observation is correctly classified as a typical point). In this work false positives and false negatives are equally undesirable as false positives may demand unnecessary and/or expensive actions for corrections and refinement and false negatives greatly reduce confidence in the data and results derived from them. The measures we considered include accuracy ($(TP+TN)/(TP+FP+FN+TN)$) which explains the overall effectiveness of a classifier; and geometric-mean (GM $= \sqrt{TP*TN}$) which explains the relative balance of TP and TN of the classifier (Sokolova & Lapalme, 2009). According to Hossin and Sulaiman (2015), these measures are not enough to capture the poor performance of the classifiers in the presence of imbalanced datasets where the size of the typical class (positive class) is much larger than the outlying class (negative class). The datasets obtained from *in situ* sensors were highly imbalanced and negatively dependent (i.e. containing many more typical observations than outliers). Therefore, we used three additional measures that are recommended for imbalanced problems with only two classes (i.e. typical and outlying) by Ranawana and Palade (2006): the negative predictive value ($NPV = TN/(FN + TN)$) which measures the probability of a negatively predicted pattern actually being negative; positive predictive value ($PPV = TP/(TP + FP)$) which measures the probability of a positively predicted pattern actually being positive; and optimized precision ($OP = P - RI$ where $P = S_pN_n + S_nN_p$; $RI = |S_p - S_n|/(S_p + S_n)$; $S_p = TN/(TN + FP)$; $S_n = TP/(TP+FN)$ and N_p and N_n represent the proportion of positives (outliers) and negatives (typical) within the entire dataset) which is a combination of accuracy, sensitivity and specificity metrics (Ranawana & Palade, 2006).

To evaluate the performance of our oddwater procedure we incorporated additional steps after detecting the outlying time points using the outlying threshold based on EVT. This was done because the time points declared as outliers by the outlying threshold could correspond to either the actual outlying points or to their neighbors. Once the time points were declared as outliers, the corresponding points in the three dimensional space were further investigated by comparing their positions with respect to the median of the typical points declared by the oddwater procedure. This step allowed us to find the most influential variable for each outlying point. For example, in Figure 2(e) the isolated point in the first quadrant is an outlier in the two dimensional space due to the outlying behavior of the conductivity measurement because the deviation of this point from the median happens primarily along the conductivity axis. In contrast, the four isolated points in the third quadrant are outliers due to the outlying behavior of the turbidity measurement because the deviations of the four points from the median happen primarily along the turbidity axis. After detecting the most influential variable for each outlying instance in the three dimensional space, further investigations were carried out separately for each individual outlying instance with respect to the most influential variable detected,to see whether the outlying instance was due to a sudden spike or a sudden drop by comparing the direction of the detected points with respect to the mean of its two immediate surrounding neighbors and itself. These additional steps in the oddwater procedure allowed us to trigger an alarm at the actual outlying point in time if the neighboring points were declared as outliers instead of the actual outliers. However, we acknowledge that these additional steps select only the most influential variable, not all of the influential variables in the presence of more than one influential variable. The additional steps were

incorporated solely to measure the performance of the oddwater procedure. In practice, and because the goal is to trigger an alarm in an occurrence of a technical outlier, it is inconsequential if the alarm is triggered either at the actual time point or at the immediate neighbouring time points corresponding to the actual outlier. As such, users of the oddwater procedure can ignore these additional steps.

Using the outlier threshold, our oddwater procedure assigns a bivariate label (either as outlier or typical point) to each observed time point and thereby creates a vector of predicted class labels. That is, if a time point is declared as an outlier by oddwater procedure, then that could be due to at least one variable in the dataset. We also declared each time point as an outlier or not based on the labels assigned by the water quality experts. At a given time point, if at least one variable was labeled as an outlier by the water quality experts then the corresponding time point was marked as an outlier and thereby creating a vector of ground-truth labels. Then the performance measures were calculated based on these two vectors of ground-truth labels and predicted class labels. Thus, this performance evaluation was done with respect to the algorithm's ability to label a point in time as an outlier or not (i.e., a point in time is an outlier if the observed value for any one or more of the three variables measured at that point in time are outliers).

2.8 Software implementation

The oddwater procedure was implemented in the open source R package `oddwater` (Talagala & Hyndman, 2018), which provides a growing list of transformation and outlier scoring methods for high dimensional data together with visualization and performance evaluation techniques. Version 0.6.0 of the package `oddwater` was used for the results presented herein and is available from Github (https://github.com/pridiltal/oddwater). In addition to the implementations available through `oddwater` package, `DDoutlier` package (Madsen, 2018) was also used for outlier score calculations. We measured the computation time (minimum (min_t), mean (mu_t), maximum (max_t) execution time) using the `microbenchmark` package (Mersmann, 2018) for different combinations of algorithms, transformations and variable combinations on 28 core Xeon-E5-2680-v4 @ 2.40GHz servers. We also developed an R Shiny application (available via `oddwater` R package) to provide interactive visual analytic tools to gain greater insight into the data and perform preliminary investigations of the relationships between water-quality variables at different sites. Further, we have archived a snapshot of version 0.6.0 of the R package on Zenodo (DOI: 10.5281/zenodo.2890469) along with the code and datasets used, with the aim of facilitating reproducibility of the results presented herein.

3 Results

3.1 Analysis of water-quality data from *in situ* sensors at Sandy Creek

A negative relationship was clearly visible between the water-quality variables: turbidity and conductivity and also between conductivity and river level measured by *in situ* sensors at Sandy Creek (Figures 3 and 4(a,c)). Further, no clear separation was observed between the target outliers and the typical points in the original data space (Figure 4(a–c)). However, a clear separation was apparent between the two sets of points once the one sided derivative transformation (an appropriate transformation for unevenly spaced data) was applied to the original series (Figures 4(d–f) and 5).

KNN-AGG and KNN-SUM algorithms performed on all three water-quality variables together, and on turbidity and conductivity together using the one sided derivative transformation, gave the highest OP (0.8329) and NPV values(0.9996), which are the most recommended measurements for negatively dependent data where the focus is

more on sensitivity (the proportion of positive patterns being correctly recognized as being positive) than specificity (Ranawana & Palade, 2006).

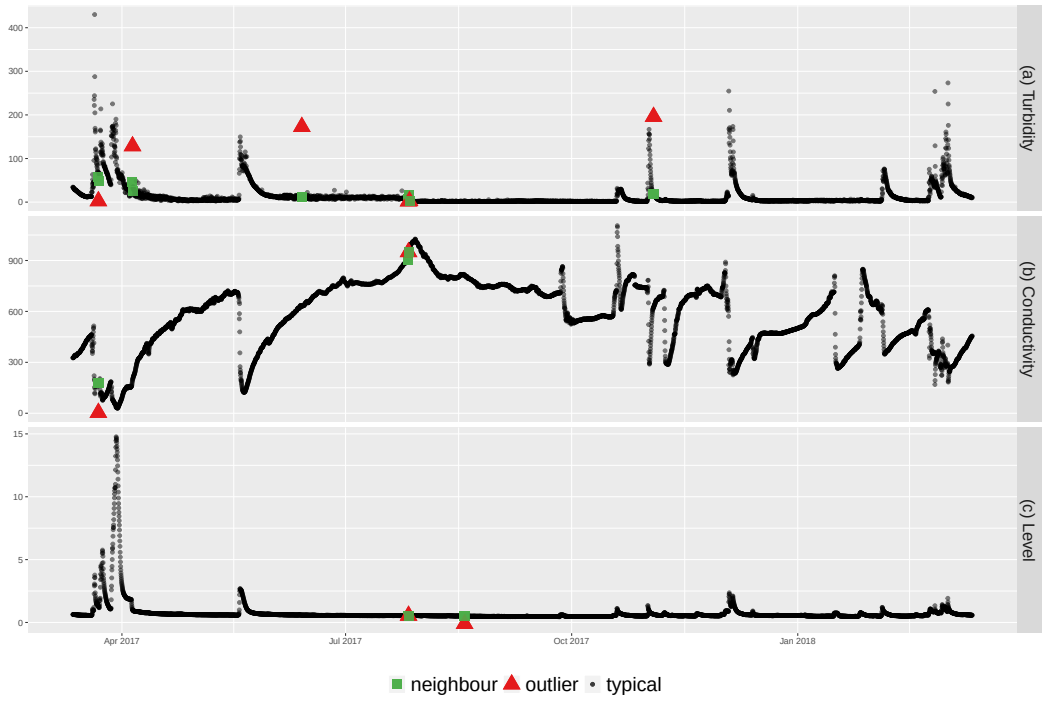

Figure 3. Time series for turbidity (NTU), conductivity (μS/cm) and river level (m) measured by *in situ* sensors at Sandy Creek. In each plot, outliers determined by water-quality experts are shown in red. Typical points are shown in black.

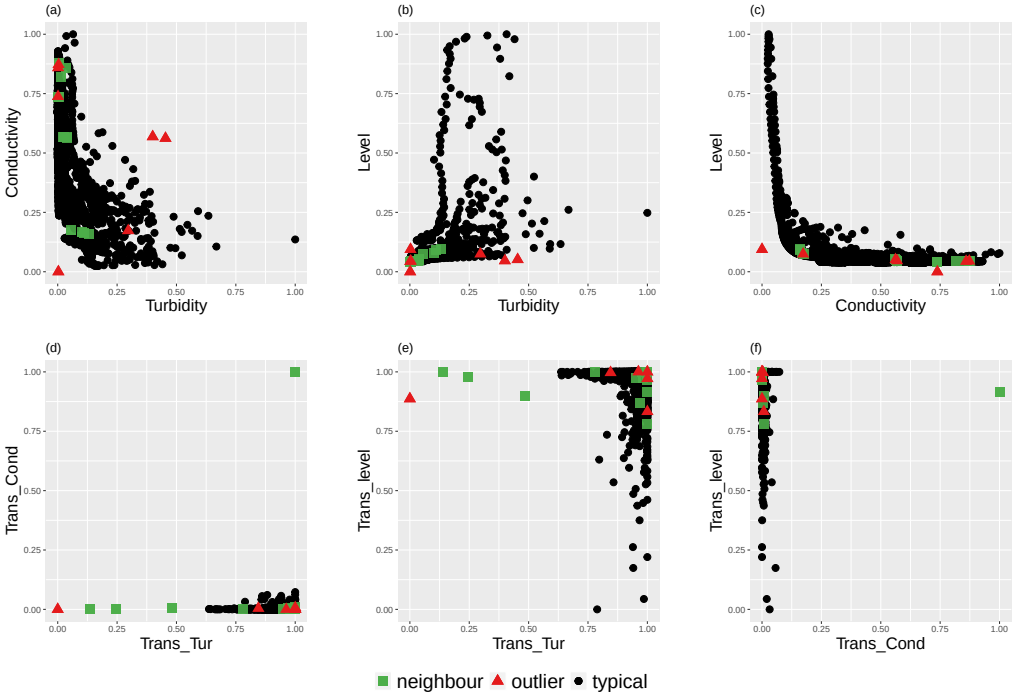

Figure 4. Top panel (a–c): Bi-variate relationships between original water-quality variables (turbidity (NTU), conductivity (μS/cm) and river level (m)) measured by *in situ* sensors at Sandy Creek. Bottom panel (d–f): Bi-variate relationships between transformed series (one sided derivative) of turbidity (NTU), conductivity (μS/cm) and river level (m) measured by *in situ* sensors at Sandy Creek. In each scatter plot, outliers determined by water-quality experts are shown in red, while typical points are shown in black. Neighboring points are marked in green.

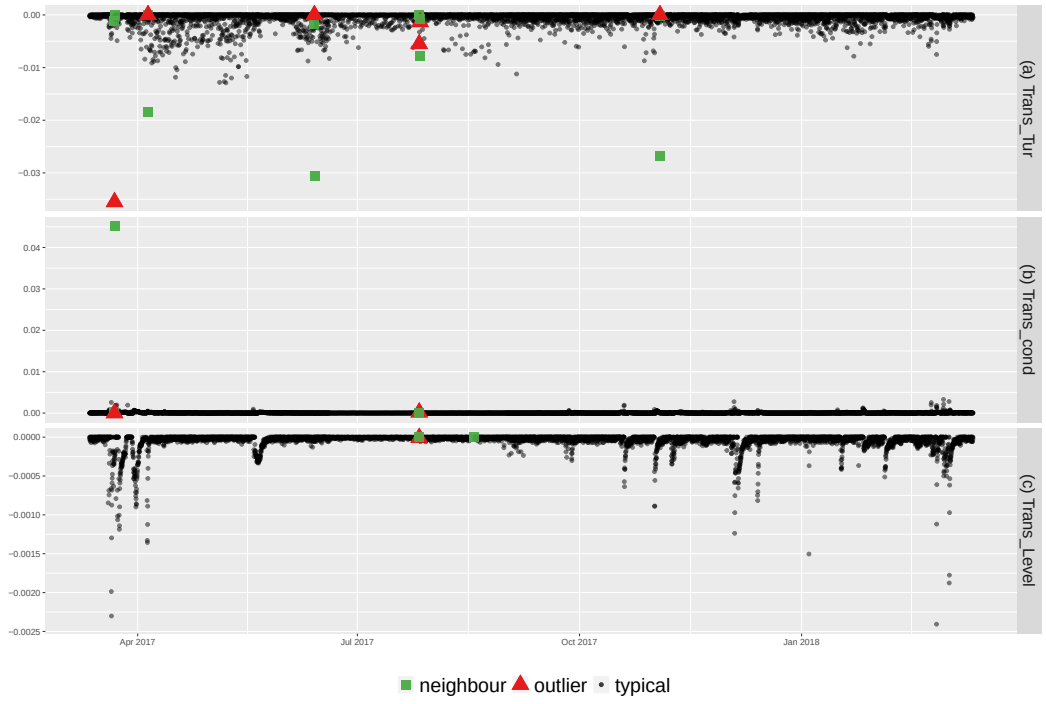

Figure 5. Transformed series (one sided derivatives) of turbidity (NTU), conductivity (μS/cm) and river level (m) measured by *in situ* sensors at Sandy Creek. In each plot outliers determined by water-quality experts are shown in red, while typical points are shown in black. Neighboring points are marked in green

Based on OP values, the one sided derivative transformation outperformed the first derivative transformation (Table 2, rows 1–5 compared to rows 6–10). Further, the distance-based outlier detection algorithms NN-HD , KNN-AGG and KNN-SUM outperformed all others (Table 2, rows 1–10 compared to rows 11–48). Among the three methods the performance of k-nearest neighbor distance-based algorithms were only slightly higher (OP = 0.8329) than the NN-HD algorithm (OP= 0.7996), which is based only on the nearest neighbor distance. The algorithm combinations with the five highest OP values also had high PPV (approximately 0.8). Furthermore, considering river level for the detection of outliers in the water-quality sensors slightly improved the performance (OP = 0.8329). Among the analysis with transformed series, LOF with the first derivative transformation performed the least well (OP= 0.2489). For most of the outlier detection algorithms (KNN-SUM, KNN-AGG, NN-HD , COF, LOF and INFLO) the poorest performances were associated with the untransformed original series, having the lowest OP and NPV values, highlighting how data transformation can improve the ability of outlier detection algorithms while maintaining low false detection rates.

Table 2. Performance metrics of outlier detection algorithms performed on multivariate water-quality time series data (T, turbidity; C, conductivity; L, river level) from in situ sensors at Sandy Creek, arranged in descending order of OP values. See Sections 2.7-8 for performance metric codes and details.

i	Variables	Transformation	Method	TN	FN	FP	TP	Accuracy	GM	OP	PPV	NPV	min_t	mu_t	max_t
1	T-C-L	One sided Derivative	KNN-AGG	5394	2	1	5	0.9994	164.23	0.83	0.83	0.9996	378.5	404.0	493.4
2	T-C-L	One sided Derivative	KNN-SUM	5394	2	1	5	0.9994	164.23	0.83	0.83	0.9996	177.2	186.8	270.3
3	T-C	First Derivative	NN-HD	5393	2	3	4	0.9991	146.87	0.80	0.57	0.9996	40.5	45.0	72.1
4	T-C	First Derivative	KNN-AGG	5392	2	4	4	0.9989	146.86	0.80	0.50	0.9996	386.3	415.8	489.4
5	T-C	One sided Derivative	NN-HD	5396	2	0	4	0.9996	146.91	0.80	1.00	0.9996	102.7	112.9	195.4
6	T-C	One sided Derivative	KNN-AGG	5395	2	1	4	0.9994	146.90	0.80	0.80	0.9996	381.8	411.7	518.0
7	T-C	One sided Derivative	KNN-SUM	5395	2	1	4	0.9994	146.90	0.80	0.80	0.9996	177.9	190.4	286.2
8	T-C-L	First Derivative	KNN-AGG	5395	4	0	3	0.9993	127.22	0.60	1.00	0.9993	377.6	404.4	476.2
9	T-C-L	First Derivative	KNN-SUM	5395	4	0	3	0.9993	127.22	0.60	1.00	0.9993	179.2	188.9	273.3
10	T-C	First Derivative	KNN-SUM	5396	4	0	2	0.9993	103.88	0.50	1.00	0.9993	179.0	189.5	283.5
11	T-C	First Derivative	LDOF	5395	4	1	2	0.9991	103.87	0.50	0.67	0.9993	17261.5	17444.7	17809.9
12	T-C	One sided Derivative	LDOF	5395	4	1	2	0.9991	103.87	0.50	0.67	0.9993	17024.3	17253.8	18079.4
13	T-C-L	First Derivative	NN-HD	5395	5	0	2	0.9991	103.87	0.44	1.00	0.9991	48.7	52.5	66.9
14	T-C-L	First Derivative	INFLO	5381	5	14	2	0.9965	103.74	0.44	0.12	0.9991	1076.5	1107.9	1168.3
15	T-C-L	First Derivative	COF	5393	5	2	2	0.9987	103.86	0.44	0.50	0.9991	5869.1	5939.8	6394.2
16	T-C-L	First Derivative	RKOF	5380	5	15	2	0.9963	103.73	0.44	0.12	0.9991	341.6	369.7	456.1
17	T-C-L	One sided Derivative	NN-HD	5395	5	0	2	0.9991	103.87	0.44	1.00	0.9991	110.4	118.2	193.2
18	T-C-L	One sided Derivative	INFLO	5392	5	3	2	0.9985	103.85	0.44	0.40	0.9991	1071.5	1113.6	1177.6
19	T-C-L	One sided Derivative	COF	5393	5	2	2	0.9987	103.86	0.44	0.50	0.9991	5676.8	5787.4	6238.4
20	T-C-L	One sided Derivative	LDOF	5392	5	3	2	0.9985	103.85	0.44	0.40	0.9991	17181.5	17261.9	17435.8
21	T-C-L	One sided Derivative	LOF	5392	5	3	2	0.9985	103.85	0.44	0.40	0.9991	500.2	516.9	596.9
22	T-C-L	One sided Derivative	RKOF	5387	5	8	2	0.9976	103.80	0.44	0.20	0.9991	338.9	370.5	464.0
23	T-C-L	Original series	KNN-AGG	5394	5	1	2	0.9989	103.87	0.44	0.67	0.9991	376.6	391.6	465.3
24	T-C-L	Original series	INFLO	5386	5	9	2	0.9974	103.79	0.44	0.18	0.9991	1034.3	1070.7	1136.7
25	T-C-L	Original series	LDOF	5393	5	2	2	0.9987	103.86	0.44	0.50	0.9991	17078.5	17156.9	17308.1
26	T-C-L	Original series	RKOF	5392	5	3	2	0.9985	103.85	0.44	0.40	0.9991	322.3	354.0	426.4
27	T-C	First Derivative	INFLO	5392	5	4	1	0.9983	73.43	0.28	0.20	0.9991	1134.6	1194.9	1271.1
28	T-C	First Derivative	COF	5396	5	0	1	0.9991	73.46	0.28	1.00	0.9991	5881.5	5991.8	6552.4
29	T-C	First Derivative	LOF	5394	5	2	1	0.9987	73.44	0.28	0.33	0.9991	498.3	512.3	596.1
30	T-C	First Derivative	RKOF	5392	5	4	1	0.9983	73.43	0.28	0.20	0.9991	335.1	363.2	435.7
31	T-C	One sided Derivative	INFLO	5394	5	2	1	0.9987	73.44	0.28	0.33	0.9991	1153.1	1207.0	1281.9
32	T-C	One sided Derivative	COF	5394	5	2	1	0.9987	73.44	0.28	0.33	0.9991	5755.0	5880.8	6420.8
33	T-C	One sided Derivative	LOF	5384	5	12	1	0.9969	73.38	0.28	0.08	0.9991	501.1	511.3	585.7
34	T-C	One sided Derivative	RKOF	5380	5	16	1	0.9961	73.35	0.28	0.06	0.9991	339.5	368.3	456.6
35	T-C	Original series	KNN-AGG	5395	5	1	1	0.9989	73.45	0.28	0.50	0.9991	371.5	405.1	483.7
36	T-C	Original series	INFLO	5387	5	9	1	0.9974	73.40	0.28	0.10	0.9991	1095.2	1143.6	1219.4
37	T-C	Original series	LDOF	5394	5	2	1	0.9987	73.44	0.28	0.33	0.9991	16842.1	17022.9	17414.4
38	T-C	Original series	RKOF	5393	5	3	1	0.9985	73.44	0.28	0.25	0.9991	321.0	351.8	440.3
39	T-C-L	First Derivative	LDOF	5395	6	0	1	0.9989	73.45	0.25	1.00	0.9989	17253.9	17323.2	17400.9
40	T-C-L	First Derivative	LOF	5395	6	0	1	0.9989	73.45	0.25	1.00	0.9989	504.6	517.1	604.4
41	T-C-L	Original series	NN-HD	5394	6	1	1	0.9987	73.44	0.25	0.50	0.9989	45.4	48.6	60.3
42	T-C-L	Original series	KNN-SUM	5395	6	0	1	0.9989	73.45	0.25	1.00	0.9989	164.7	177.3	243.4
43	T-C-L	Original series	COF	5395	6	0	1	0.9989	73.45	0.25	1.00	0.9989	5864.4	5931.7	6329.7
44	T-C-L	Original series	LOF	5395	6	0	1	0.9989	73.45	0.25	1.00	0.9989	480.2	505.0	576.2
45	T-C	Original series	NN-HD	5395	6	1	0	0.9987	0.00	0.00	0.00	0.9989	38.1	41.7	66.3
46	T-C	Original series	KNN-SUM	5396	6	0	0	0.9989	0.00	0.00	NaN	0.9989	172.7	184.6	272.5
47	T-C	Original series	COF	5396	6	0	0	0.9989	0.00	0.00	NaN	0.9989	5826.3	5896.4	6804.3
48	T-C	Original series	LOF	5396	6	0	0	0.9989	0.00	0.00	NaN	0.9989	477.0	502.7	568.0

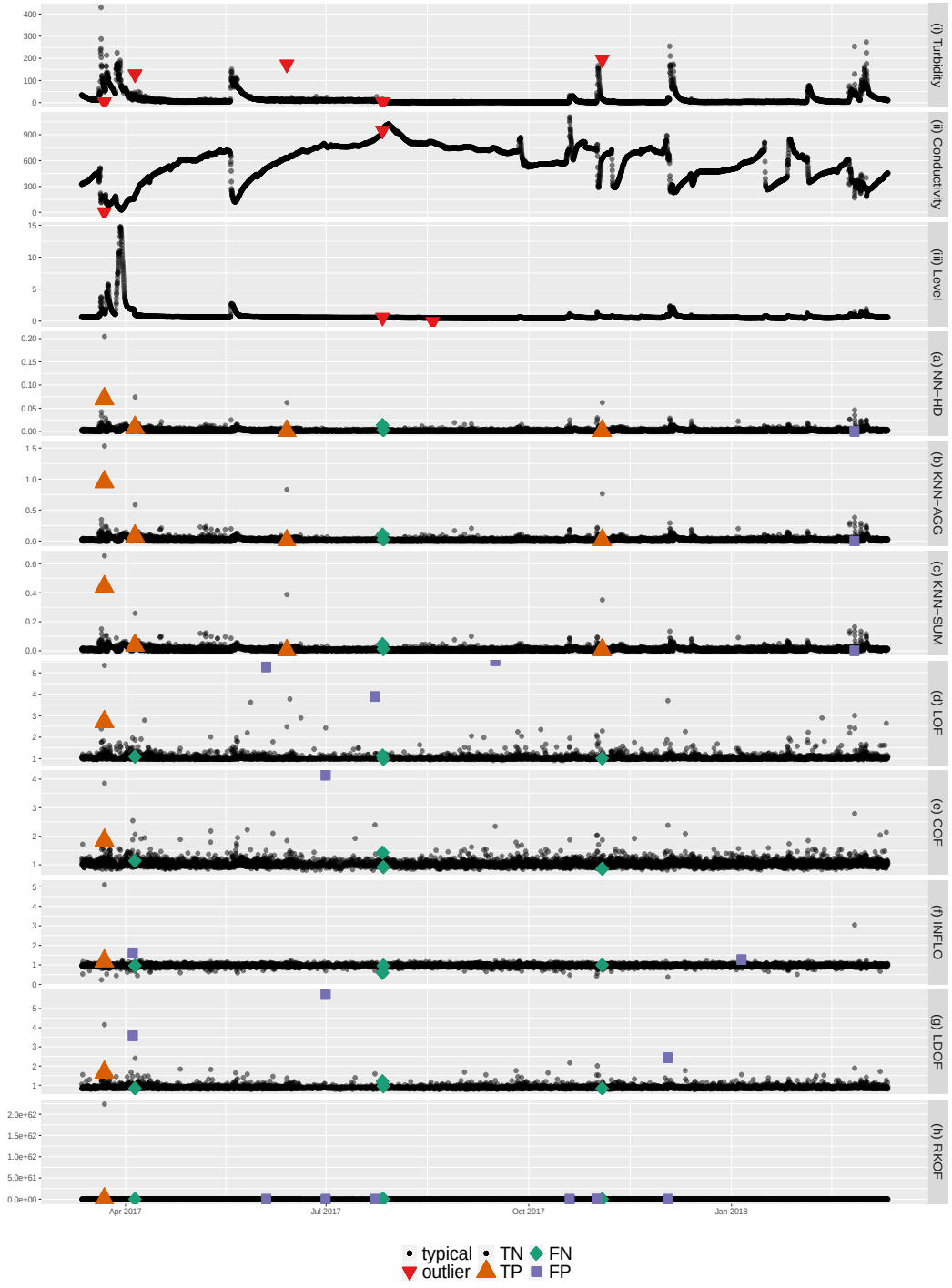

Figure 6. Classification of outlier scores produced from different algorithms as true negatives (TN), true positives (TP), false negatives (FN), false positives (FP). The top three panels (i, ii, iii) correspond to the original series (turbidity, conductivity and river level) measured by *in situ* sensors at Sandy Creek. The target outliers (detected by water-quality experts) are shown in red, while typical points are shown in black. The remaining panels (a–h) give outlier scores produced by different outlier detection algorithms for high dimensional data when applied to the transformed series (one sided derivative) of the three variables: turbidity, conductivity and level. Through different outlier scoring algorithms (Panel a - h) we are evaluating whether each point in time is an outlier or not. Therefore, from Panel a-h, if the outlier scoring algorithm is effective, then there should be either TP or TN at each point in time when either a red triangle is plotted in at least one of the three panels (i- iii), or black dots are plotted in all of the top three panels (i - iii), respectively. Since outlier scores are non negative and are mostly clustered near zero, with some occasional high values, a square root transformation was applied to reduce skewness of the data in Panel (a) to (h).

The three outlier detection algorithms that demonstrated the highest level of accuracy (NN-HD, KNN-AGG and KNN-SUM) also outperformed the others with respect to computational time. NN-HD algorithm required the least computational time. Among the remaining two, the mean computational time of KNN-AGG ($\approx$ 400 milliseconds) was twice that of KNN-SUM's ($<$ 200 milliseconds). LOF and its extensions (INFLO, COF and LDOF) demonstrated the poorest performance with respect computational time ($>$ 500 milliseconds on average).

Only KNN-SUM and KNN-AGG assigned high scores to most of the targeted outliers in turbidity, conductivity and level data transformed using the one-sided derivative (Figure 6(a,b)). For each outlying instance however, the next immediate neighboring point was assigned the high outlier score instead of the true outlying point. After determining the most influential variable using the additional steps of the algorithm (Section 2.7), adjustments were made to correct this to the actual outlier. The outlier scores produced by LOF and COF (Figure 6(d,f)) were unable to capture the outlying behaviors correctly and demonstrated high scattering. In comparison to other outlier scoring algorithms, KNN-SUM algorithm displayed a good compromise between accuracy and computational efficiency (Table 2).

3.2 Analysis of water-quality data from *in situ* sensors at Pioneer River

Some of the target outliers in the data obtained from the *in situ* sensors at Pioneer River only deviated slightly from the general trend (Figure 7), making outlier detection challenging. A negative relationship was clearly visible between turbidity and conductivity (Figure 8(a)), however the relationship between level and conductivity was complex (Figure 8(c)). Most of the target outliers were masked by the typical points in the original space (Figure 8(a–c)). Similar to Sandy Creek, data obtained from the sensors at Pioneer River showed good separation between outliers and typical points under the one sided derivative transformation (Figures 8(d–f) and 9). However, the sudden spikes in turbidity labeled as outliers by water-quality experts could not be separated from the majority by a large distance and were only visible as a small group (micro cluster (Goldstein & Uchida, 2016)) in the boundary defined by the typical points (Figure 8(d, e)).

From the performance analysis it was observed that turbidity and conductivity together produced better results (Table 3, rows 1–3) than when combined with river level, which tended to reduce the performance (i.e. generating lower OP and NPV values) while increasing the false negative rate (Table 3, rows 4–5). KNN-AGG and KNN-SUM (Table 3, rows 1–3) had the highest accuracy (0.9978), lowest error rates (0.0022), highest geometric means (492.8012), highest OP (0.8845) and highest NPV (0.9984). Despite the challenge given by the small spikes which could not be clearly separated from the typical points, KNN-AGG, KNN-SUM and NN-HD with one sided derivatives of turbidity and conductivity still detected some of those points as outliers while maintaining low false negative and false positive rates. Similar to Sandy Creek, NN-HD ($<$ 200 milliseconds on average) and KNN-SUM ($<$ 230 milliseconds on average) demonstrated the highest computational efficiency for the data obtained from Pioneer River.

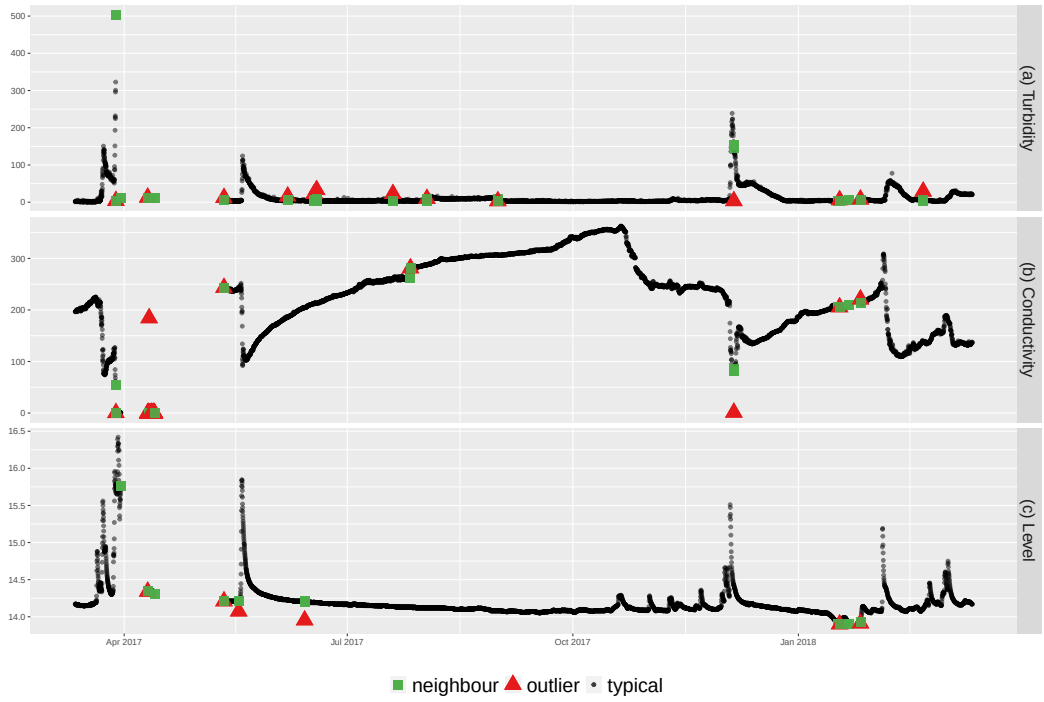

Figure 7. Time series for turbidity (NTU), conductivity (μS/cm) and river level (m) measured by *in situ* sensors at Pioneer River. In each plot, outliers determined by water-quality experts are shown in red, while typical points are shown in black.

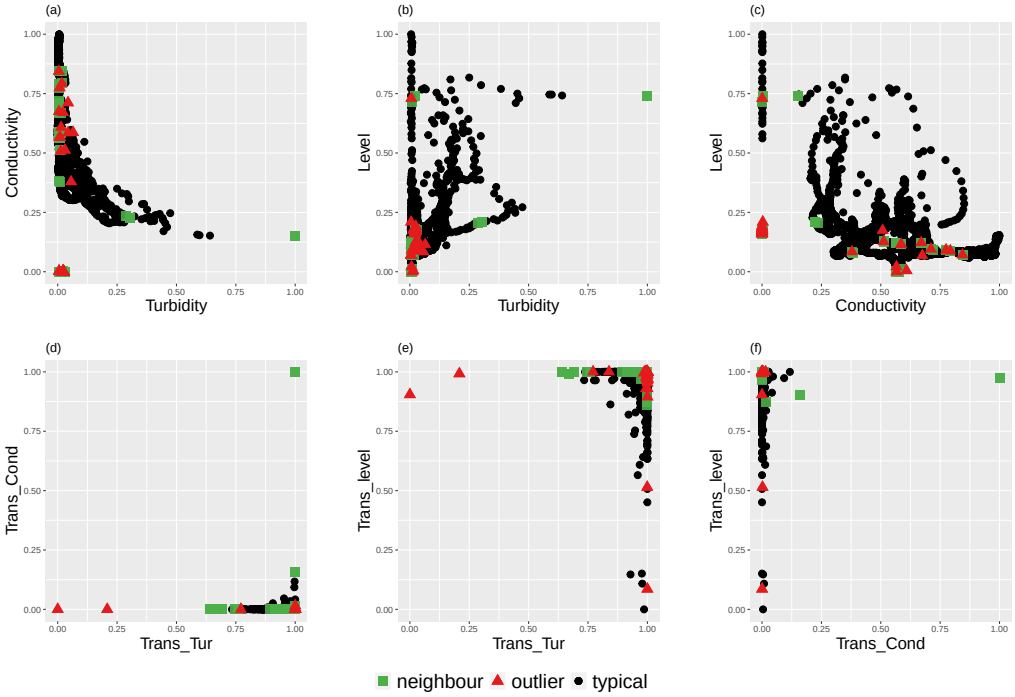

Figure 8. Top panel (a–c): Bi-variate relationships between original water-quality variables (turbidity (NTU), conductivity (μS/cm) and river level (m)) measured by *in situ* sensors at Pioneer River. Bottom panel (d–f): Bi-variate relationships between transformed series (one sided derivative) of turbidity (NTU), conductivity (μS/cm) and river level (m) measured by *in situ* sensors at Pioneer River. In each scatter plot, outliers determined by water-quality experts are shown in red, while typical points are shown in black. Neighboring points are marked in green.

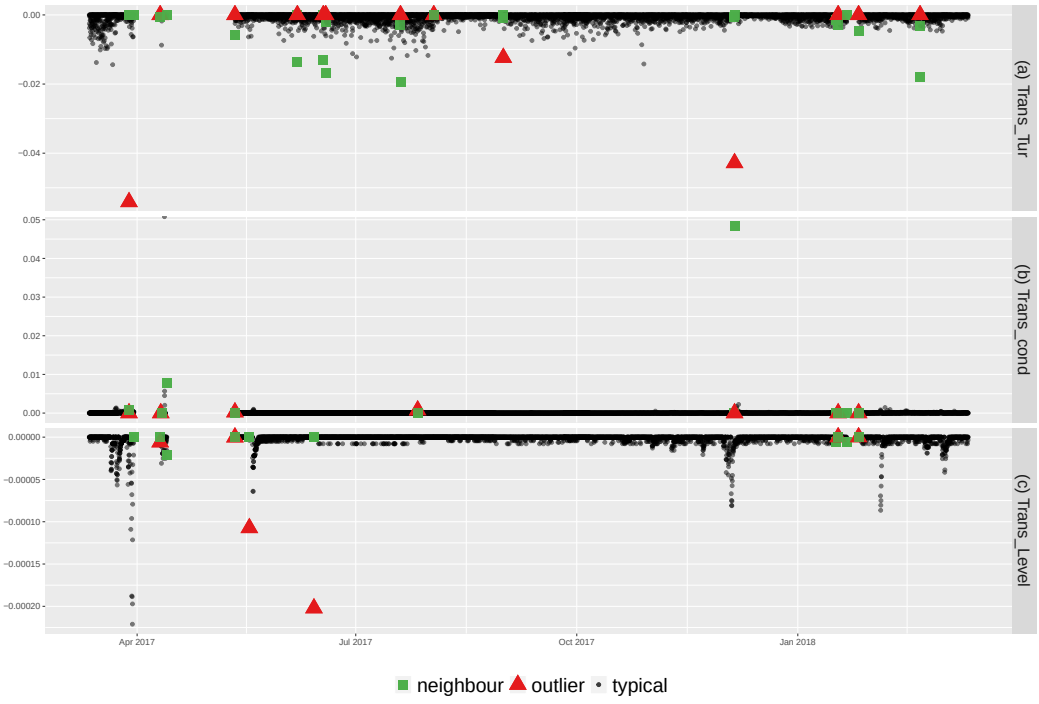

Figure 9. Transformed series (one sided derivatives) of turbidity (NTU), conductivity (μS/cm) and river level (m) measured by *in situ* sensors at Pioneer River. In each plot, outliers determined by water-quality experts are shown in red, while typical points are shown in black. Neighboring points are marked in green

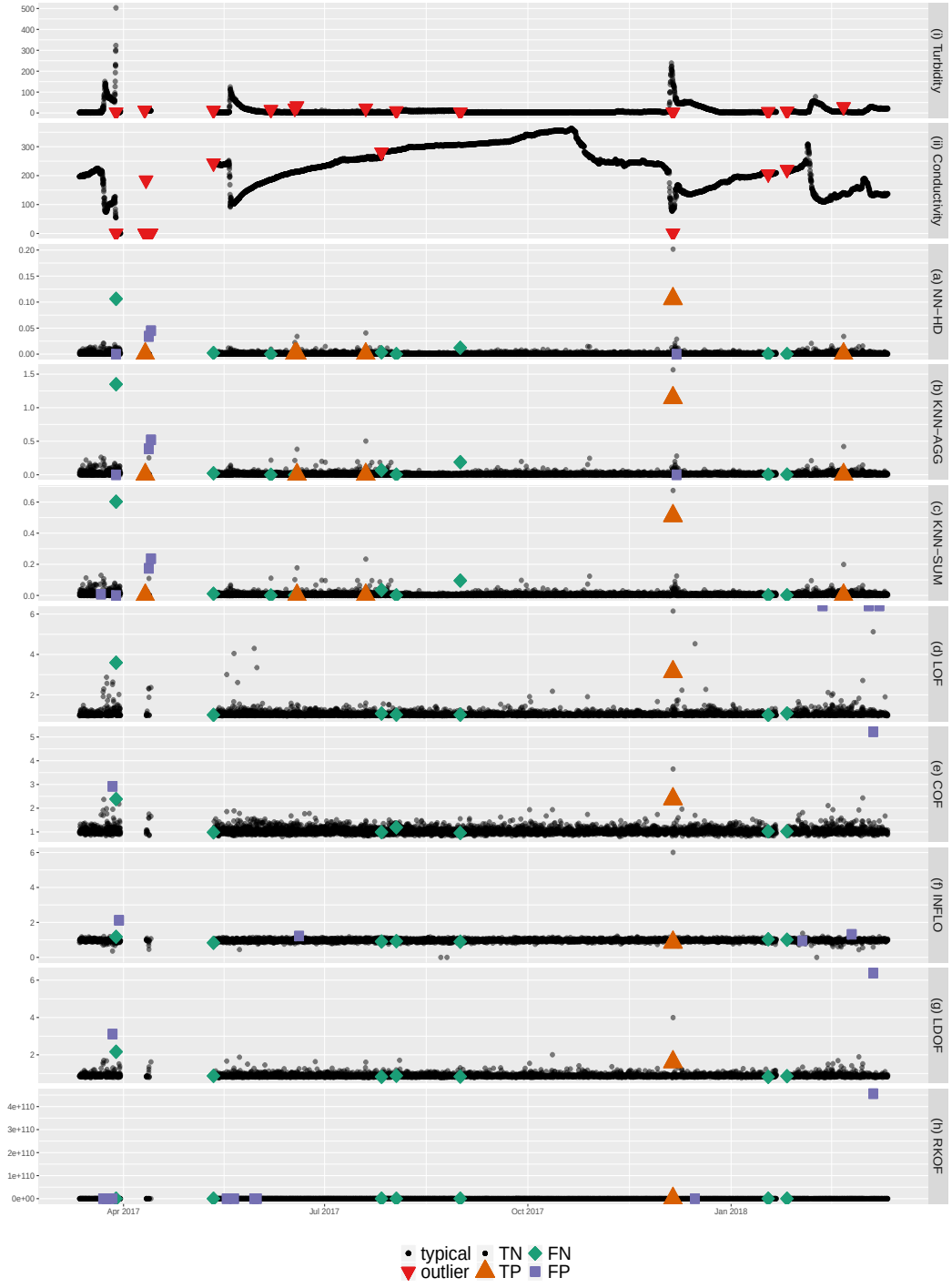

Figure 10. Classification of outlier scores produced from different algorithms as true negatives (TN), true positives (TP), false negatives (FN), false positives (FP). The top two panels (i and ii) correspond to the original series (turbidity and conductivity) measured by *in situ* sensors at Pioneer River. The target outliers (detected by water-quality experts) are shown in red, while typical points are shown in black. The remaining panels (a–h) give outlier scores produced by different outlier detection algorithms for high dimensional data when applied to the transformed series (one sided derivative) of the two variables: turbidity and conductivity. Through different outlier scoring algorithms (Panel a - h) we are evaluating whether each point in time is an outlier or not. Therefore, from Panel a-h, if the outlier scoring algorithm is effective, then there should be either TP or TN at each point in time when either a red triangle is plotted in at least one of the two panels (i- ii), or black dots are plotted in both of the top two panels (i - ii), respectively. Since outlier scores are non negative and are mostly clustered near zero, with some occasional high values, a square root transformation was applied to reduce skewness of the data in Panel (a) to (h).

Table 3. Performance metrics of outlier detection algorithms performed on multivariate water-quality time series data (T, turbidity; C, conductivity; L, river level) from in situ sensors at Pioneer River, arranged in descending order of OP values. See Sections 2.7-8 for performance metric codes and details.

i	Variables	Transformation	Method	TN	FN	FP	TP	Accuracy	GM	OP	PPV	NPV	min_t	mu_t	max_t
1	T-C	One sided Derivative	NN-HD	6226	10	5	39	0.9976	492.76	0.88	0.89	0.9984	128.0	136.5	257.7
2	T-C	One sided Derivative	KNN-AGG	6227	10	4	39	0.9978	492.80	0.88	0.91	0.9984	443.5	478.8	564.6
3	T-C	One sided Derivative	KNN-SUM	6227	10	4	39	0.9978	492.80	0.88	0.91	0.9984	209.6	222.2	325.5
4	T-C	First Derivative	NN-HD	6229	12	2	37	0.9978	480.08	0.86	0.95	0.9981	169.6	182.0	272.3
5	T-C	First Derivative	KNN-AGG	6229	12	2	37	0.9978	480.08	0.86	0.95	0.9981	449.5	488.5	588.2
6	T-C	First Derivative	KNN-SUM	6229	12	2	37	0.9978	480.08	0.86	0.95	0.9981	212.1	225.3	325.9
7	T-C	First Derivative	INFLO	6225	12	6	37	0.9971	479.92	0.86	0.86	0.9981	1452.1	1525.0	1613.2
8	T-C	First Derivative	RKOF	6224	12	7	37	0.9970	479.88	0.86	0.84	0.9981	400.2	430.4	523.9
9	T-C-L	One sided Derivative	KNN-AGG	6225	12	4	39	0.9975	492.72	0.86	0.91	0.9981	437.4	465.2	541.6
10	T-C-L	One sided Derivative	KNN-SUM	6225	12	4	39	0.9975	492.72	0.86	0.91	0.9981	195.6	214.5	297.8
11	T-C-L	First Derivative	RKOF	6211	13	18	38	0.9951	485.82	0.85	0.68	0.9979	396.9	425.9	503.4
12	T-C-L	First Derivative	KNN-AGG	6227	14	2	37	0.9975	480.00	0.84	0.95	0.9978	460.8	478.0	570.3
13	T-C-L	First Derivative	KNN-SUM	6227	14	2	37	0.9975	480.00	0.84	0.95	0.9978	201.5	220.0	292.2
14	T-C	First Derivative	COF	6230	13	1	36	0.9978	473.58	0.84	0.97	0.9979	7812.7	7908.2	8453.3
15	T-C	First Derivative	LDOF	6230	13	1	36	0.9978	473.58	0.84	0.97	0.9979	23241.0	23435.7	24522.1
16	T-C	First Derivative	LOF	6228	13	3	36	0.9975	473.51	0.84	0.92	0.9979	562.6	594.4	668.3
17	T-C	One sided Derivative	INFLO	6227	13	4	36	0.9973	473.47	0.84	0.90	0.9979	1488.8	1559.9	1633.1
18	T-C	One sided Derivative	COF	6229	13	2	36	0.9976	473.54	0.84	0.95	0.9979	7393.6	7505.5	8037.1
19	T-C	One sided Derivative	LDOF	6228	13	3	36	0.9975	473.51	0.84	0.92	0.9979	22802.2	22986.0	23561.3
20	T-C	One sided Derivative	LOF	6228	13	3	36	0.9975	473.51	0.84	0.92	0.9979	581.9	596.9	682.6
21	T-C	One sided Derivative	RKOF	6219	13	12	36	0.9960	473.16	0.84	0.75	0.9979	388.6	419.7	510.4
22	T-C	Original Series	INFLO	6227	13	4	36	0.9973	473.47	0.84	0.90	0.9979	1405.6	1498.5	1578.2
23	T-C-L	First Derivative	COF	6228	15	1	36	0.9975	473.51	0.83	0.97	0.9976	7823.0	7910.7	8344.7
24	T-C-L	First Derivative	LDOF	6228	15	1	36	0.9975	473.51	0.83	0.97	0.9976	23220.1	23357.7	23878.3
25	T-C-L	One sided Derivative	NN-HD	6228	15	1	36	0.9975	473.51	0.83	0.97	0.9976	125.7	131.9	206.1
26	T-C	Original Series	NN-HD	6230	14	1	35	0.9976	466.96	0.83	0.97	0.9978	159.9	171.0	278.2
27	T-C	Original Series	KNN-AGG	6226	14	5	35	0.9970	466.81	0.83	0.88	0.9978	434.2	468.7	553.0
28	T-C	Original Series	KNN-SUM	6226	14	5	35	0.9970	466.81	0.83	0.88	0.9978	192.9	211.6	305.8
29	T-C	Original Series	COF	6231	14	0	35	0.9978	467.00	0.83	1.00	0.9978	7518.5	7617.6	8501.1
30	T-C	Original Series	LDOF	6231	14	0	35	0.9978	467.00	0.83	1.00	0.9978	22770.9	22910.4	23857.1
31	T-C	Original Series	LOF	6231	14	0	35	0.9978	467.00	0.83	1.00	0.9978	551.2	579.1	632.6
32	T-C	Original Series	RKOF	6222	14	9	35	0.9963	466.66	0.83	0.80	0.9978	373.6	401.9	475.4
33	T-C-L	First Derivative	NN-HD	6227	15	2	36	0.9973	473.47	0.82	0.95	0.9976	157.3	167.1	244.3
34	T-C-L	One sided Derivative	INFLO	6226	15	3	36	0.9971	473.43	0.82	0.92	0.9976	1383.5	1418.8	1477.4
35	T-C-L	One sided Derivative	COF	6227	15	2	36	0.9973	473.47	0.82	0.95	0.9976	7414.6	7497.9	7899.6
36	T-C-L	One sided Derivative	LDOF	6227	15	2	36	0.9973	473.47	0.82	0.95	0.9976	22756.8	23090.7	23941.1
37	T-C-L	One sided Derivative	RKOF	6214	15	15	36	0.9952	472.97	0.82	0.71	0.9976	390.5	422.1	490.3
38	T-C-L	First Derivative	INFLO	6229	16	0	35	0.9975	466.92	0.81	1.00	0.9974	1344.7	1398.3	1456.9
39	T-C-L	First Derivative	LOF	6229	16	0	35	0.9975	466.92	0.81	1.00	0.9974	585.6	600.7	688.1
40	T-C-L	One sided Derivative	LOF	6223	16	6	35	0.9965	466.70	0.81	0.85	0.9974	583.4	596.1	672.1
41	T-C-L	Original Series	NN-HD	6228	16	1	35	0.9973	466.88	0.81	0.97	0.9974	152.8	163.0	231.2
42	T-C-L	Original Series	KNN-AGG	6224	16	5	35	0.9967	466.73	0.81	0.88	0.9974	439.3	456.3	534.2
43	T-C-L	Original Series	KNN-SUM	6224	16	5	35	0.9967	466.73	0.81	0.88	0.9974	186.5	201.4	269.6
44	T-C-L	Original Series	INFLO	6229	16	0	35	0.9975	466.92	0.81	1.00	0.9974	1329.9	1372.8	1415.0
45	T-C-L	Original Series	COF	6229	16	0	35	0.9975	466.92	0.81	1.00	0.9974	7596.5	7707.2	8357.8
46	T-C-L	Original Series	LDOF	6229	16	0	35	0.9975	466.92	0.81	1.00	0.9974	22897.7	127337.1	10458496.0
47	T-C-L	Original Series	LOF	6229	16	0	35	0.9975	466.92	0.81	1.00	0.9974	549.5	580.9	646.9
48	T-C-L	Original Series	RKOF	6217	16	12	35	0.9955	466.47	0.81	0.74	0.9974	368.3	406.8	497.2

4 Discussion

We introduced a new procedure, named oddwater procedure for the detection of outliers in water-quality data from *in situ* sensors, where outliers were specifically defined as due to technical errors that make the data unreliable and untrustworthy. We showed that our oddwater procedure, with carefully selected data transformation methods derived from data features, can greatly assist in increasing the performance of a range of existing outlier detection algorithms. Our oddwater procedure, and analysis using data obtained from *in situ* sensors positioned at two study sites, Sandy Creek and Pioneer River, performed well with outlier types such as sudden isolated spikes, sudden isolated drops, and level shifts, while maintaining low false detection rates. As an unsupervised procedure, our approach can be easily extended to other water-quality variables, other sites and also to other outlier detection tasks in other application domains. The only requirement is to select suitable transformation methods according to the data features that differentiate the outlying instances from the typical behaviors of a given system.

Studies have shown that transforming variables affects densities, relative distances and orientation of points within the data space and therefore can improve the ability to perceive patterns in the data which are not clearly visible in the original data space (Dang & Wilkinson, 2014). This was the case in our study, where no clear separation was visible between outliers and typical data points in the original data space, but a clear separation was obtained between the two sets of points once the one-sided derivative transformation was applied to the original series. Having this type of a separation between outliers and typical points is important before applying unsupervised outlier detection algorithms for high dimensional data because the methods are usually based on the definition of outliers in terms of distance or density (Talagala et al., 2018). Most of the outlier detection algorithms (KNN-SUM, KNN-AGG, NN-HD , COF, LOF and INFLO) performed least well with the untransformed original series, demonstrating how data transformation methods can assist in improving the ability of outlier detection algorithms while maintaining low false detection rates.

Although outlying points were clearly separated from their majority, which corresponded to the typical behaviors, the individual outliers were not isolated and were surrounded by the other outlying points. Because NN-HD has the additional requirement of isolation in addition to clear separation between outlying points and typical points, it performed poorly in comparison to the two KNN distance-based algorithms (KNN-AGG and KNN-SUM) which are not restricted to the single most nearest neighbor (Talagala et al., 2018). For the current work k was set to 10, the maximum default value of k in Madsen (2018), because too large a value of k could skew the focus towards global outliers (points that deviates significantly from the rest of the dataset) alone (Zhang et al., 2009) and make the algorithms computationally inefficient. On the other hand, too small a value of k could incorporate an additional assumption of isolation into the algorithm, as in the NN-HD algorithm where $k = 1$. Among the analysis using transformed series, LOF with the first derivative transformation performed the least well, which could also be due to its additional assumption of isolation (Tang et al., 2002). However, using the same k across all algorithms may bias direct comparison as the performance of the algorithms can depend on the value of k and algorithms can reach their peak performance for different choices of k (Campos et al., 2016). Therefore performing an optimisation to select the best k is non trivial and we leave it for future work.

We took the correlation structure between the variables into account when detecting outliers as some were apparent only in the high dimensional space but not when each variable was considered independently (Ben-Gal, 2005). A negative relationship was observed between conductivity and turbidity and also between conductivity and level for the Sandy Creek data. However, for Pioneer River, no clear relationship was observed between level and the remaining two variables, turbidity and conductivity. This could be one reason why the variable combination with river level gave poor results for the Pioneer River dataset, while results for other combinations were similar to those of Sandy Creek.

The one-sided derivative transformation outperformed the derivative transformation. This was expected because in an occurrence of a sudden spike or isolated drop the first derivative assigns high values to two consecutive points, the actual outlying point as well as the neighboring point, and therefore increases the false positive rate (because the neighboring points that are declared to be outliers actually correspond to typical points in the original data space).

Our goal was to detect suitable transformations, combinations of variables, and the algorithms for outlier score calculation for the data from two study sites. Results may depend on the characteristics of the time series (site and time dependent for example), and what is best for one site may not be the best for another site. Therefore care should be taken to select transformations most suitable for the problem at hand. According to Dang and Wilkinson (2014), any transformation used on a dataset must be evaluated

in terms of a figure of merit (i.e. a numerical quantity used to characterize the performance of a method, relative to its alternatives). For our work on detecting outliers, the figure of merit was the maximum separability of the two classes generated by outliers and typical points. However, we acknowledge that the set of transformations that we used for this work was relatively limited and influenced by the data obtained from the two study sites. Therefore, the set of transformations we considered (Table 1) should be viewed only as an illustration of our oddwater procedure for detecting outliers. We expect that the set of transformations will expand over time as the oddwater procedure is used for other data from other study sites and for applications to other fields.

For the current work we selected transformation methods that could highlight abrupt changes in the water-quality data.We hope to expand the ability of oddwater procedure so that it can detect other outlier types not previously targeted but commonly observed in water quality data (e.g.: low/high variability, drift etc. as per Leigh et al. (2019)). One possibility is to consider the residuals at each point, defined as the difference between the actual values and the fitted values (similar to Schwarz (2008)) or the difference between the actual values and the predicted values (similar to Hill and Minsker (2006)), as a transformation and apply outlier detection algorithms to the high dimensional space defined by the residuals. Here the challenge will be to identify the appropriate curve fitting and prediction models to generate the residual series. In this way, continuous subsequences of high values could correspond to other kinds of technical outliers such as high variability or drift. However, the range of applications and the space of the transformations are extremely diverse, which makes it challenging to provide a structured formal vision that covers all of the possible transformations that could be considered. The transformations we presented in this paper were mainly chosen as appropriate to the data collected from Sandy Creek and Pioneer River. We observed that different transformations can lead to entirely different data structures and that the selection of suitable transformations is directed by the data features and typical patterns imposed by a given application. Domain specific knowledge plays a vital role when selecting suitable transformations and as such defining structured guidelines for the selection of suitable transformations remains problematic.

Not surprisingly, NN-HD algorithm required the least computational time given the outlying score calculation only involves searching for the single most nearest neighbors of each test point (Wilkinson, 2018). The mean computational time of KNN-AGG was twice as high as that of KNN-SUM because the KNN-AGG algorithm has the additional requirement of calculating weights that assign nearest neighbors higher weight relative to the neighbors farther apart (Angiulli & Pizzuti, 2002). LOF and its extensions (INFLO, COF and LDOF) required the most computational time; all four algorithms involve a two step searching mechanism at each test point when calculating the corresponding outlying score. This means that at each test point each algorithm searches its k nearest neighbors as well those of the detected nearest neighbors for the outlier score calculation (Breunig et al., 2000; Tang et al., 2002; Jin et al., 2006; Zhang et al., 2009).

We hope to extend our multivariate outlier detection framework into space and time so that it can deal with the spatio-temporal correlation structure along branching river networks. Further, in the current paper we have introduced our oddwater procedure as a batch method. However, due to the unsupervised nature of our oddwater procedure it can be easily extended to a streaming data scenario with the help of a sliding window of fixed length. A streaming data scenario always demands a near-real-time support. Therefore one significant challenge is to find efficient methods that allow us to update outlier scores taking account of the newest observations and removing the oldest observations introduced by overlapping sliding windows, rather than recalculating scores corresponding to observations which are not affected by either new arrivals or the oldest observations (that are no longer covered by the latest window). Further work will be needed to

investigate the efficient computation of regenerating nearest neighbours in a data streaming context.

Acknowledgments

Funding for this project was provided by the Queensland Department of Environment and Science (DES) and the ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS). The authors would like to acknowledge the Queensland Department of Environment and Science; in particular, the Great Barrier Reef Catchment Loads Monitoring Program for the data, and the staff from Water Quality and Investigations for their input. We thank Ryan S. Turner and Erin E. Peterson for several valuable discussions regarding project requirements and water quality characteristics. Further, this research was supported in part by the Monash eResearch Centre and eSolutions-Research Support Services through the use of the MonARCH (Monash Advanced Research Computing Hybrid) HPC Cluster. We would also like to thank David Hill and other anonymous reviewers for their valuable comments and suggestions. The datasets used for this article are available in the open source R package `oddwater` (Talagala & Hyndman, 2018).

References

Angiulli, F., & Pizzuti, C. (2002). Fast outlier detection in high dimensional spaces. In *European conference on principles of data mining and knowledge discovery* (pp. 15–27).

Archer, C., Baptista, A., & Leen, T. K. (2003). Fault detection for salinity sensors in the columbia estuary. *Water Resources Research*, *39*(3).

Ben-Gal, I. (2005). Outlier detection. In *Data mining and knowledge discovery handbook* (pp. 131–146). Springer.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). Lof: identifying density-based local outliers. In *Acm sigmod record* (Vol. 29, pp. 93–104).

Burridge, P., & Taylor, A. M. R. (2006). Additive outlier detection via extreme-value theory. *Journal of Time Series Analysis*, *27*(5), 685–701.

Campos, G. O., Zimek, A., Sander, J., Campello, R. J., Micenková, B., Schubert, E., ... Houle, M. E. (2016). On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery*, *30*(4), 891–927.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, *41*(3), 15.

Dang, T. N., & Wilkinson, L. (2014). Transforming scagnostics to reveal hidden features. *IEEE transactions on visualization and computer graphics*, *20*(12), 1624–1632.

Embrechts, P., Klüppelberg, C., & Mikosch, T. (2013). *Modelling extremal events: for insurance and finance*. Springer Berlin Heidelberg. Retrieved from `https://books.google.com.au/books?id=BXOI2pICfJUC`

Gao, J., Hu, W., Zhang, Z. M., Zhang, X., & Wu, O. (2011). Rkof: robust kernel-based local outlier detection. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 270–283).

Glasgow, H. B., Burkholder, J. M., Reed, R. E., Lewitus, A. J., & Kleinman, J. E. (2004). Real-time remote monitoring of water quality: a review of current applications, and advancements in sensor, telemetry, and computing technologies. *Journal of Experimental Marine Biology and Ecology*, *300*(1-2), 409–448.

Goldstein, M., & Uchida, S. (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, *11*(4), e0152173.

Hill, D. J., & Minsker, B. S. (2006). Automated fault detection for in-situ environmental sensors. In *Proceedings of the 7th international conference on hydroinformatics.*

Hill, D. J., Minsker, B. S., & Amir, E. (2009). Real-time bayesian anomaly detection in streaming environmental data. *Water Resources Research*, *45*(4).

Hossin, M., & Sulaiman, M. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, *5*(2), 1.

Jin, W., Tung, A. K., Han, J., & Wang, W. (2006). Ranking outliers using symmetric neighborhood relationship. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 577–593).

Koch, M. W., & McKenna, S. A. (2010). Distributed sensor fusion in water quality event detection. *Journal of Water Resources Planning and Management*, *137*(1), 10–19.

Kotamäki, N., Thessler, S., Koskiaho, J., Hannukkala, A. O., Huitu, H., Huttula, T., ... Järvenpää, M. (2009). Wireless in-situ sensor network for agriculture and water monitoring on a river basin scale in southern finland: Evaluation from a data users perspective. *Sensors*, *9*(4), 2862–2883.

Kriegel, H.-P., Kröger, P., & Zimek, A. (2010). Outlier detection techniques. *Tutorial at KDD*, *10*.

Leigh, C., Alsibai, O., Hyndman, R. J., Kandanaarachchi, S., King, O. C., McGree, J. M., ... others (2019). A framework for automated anomaly detection in high frequency water-quality data from in situ sensors. *Science of The Total Environment*, *664*, 885–898.

Madsen, J. H. (2018). Ddoutlier: Distance and density-based outlier detection [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=DDoutlier` (R package version 0.1.0)

McInnes, K., Abbs, D., Bhend, J., Chiew, F., Church, J., Ekstrm, M., ... Whetton, P. (2015). *Wet tropics cluster report: Climate change in australia projections for australia's nrm regions.* CSIRO.

McKenna, S. A., Hart, D., Klise, K., Cruz, V., & Wilson, M. (2007). Event detection from water quality time series. In *World environmental and water resources congress 2007: Restoring our natural habitat* (pp. 1–12).

Mersmann, O. (2018). microbenchmark: Accurate timing functions [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=microbenchmark` (R package version 1.4-4)

Mitchell, C., Brodie, J., & White, I. (2005). Sediments, nutrients and pesticide residues in event flow conditions in streams of the mackay whitsunday region, australia. *Marine Pollution Bulletin*, *51*(1-4), 23–36.

Moatar, F., Fessant, F., & Poirel, A. (1999). ph modelling by neural networks. application of control and validation data series in the middle loire river. *Ecological Modelling*, *120*(2-3), 141–156.

Moatar, F., Miquel, J., & Poirel, A. (2001). A quality-control method for physical and chemical monitoring data. application to dissolved oxygen levels in the river loire (france). *Journal of Hydrology*, *252*(1-4), 25–36.

Panguluri, S., Meiners, G., Hall, J., & Szabo, J. (2009). Distribution system water quality monitoring: Sensor technology evaluation methodology and results. *US Environ. Protection Agency, Washington, DC, USA, Tech. Rep. EPA/600/R-09/076*, *2772*.

R Core Team. (2018). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from `https://www.R-project.org/`

Raciti, M., Cucurull, J., & Nadjm-Tehrani, S. (2012). Anomaly detection in water management systems. In *Critical infrastructure protection* (pp. 98–119). Springer.

Ranawana, R., & Palade, V. (2006). Optimized precision-a new measure for classifier performance evaluation. In *Evolutionary computation, 2006. cec 2006. ieee congress on* (pp. 2254–2261).

Rangeti, I., Dzwairo, B., Barratt, G. J., & Otieno, F. A. (2015). Validity and errors in water quality dataa review. In *Research and practices in water quality.* In-Tech.

Schwarz, K. T. (2008). *Wind dispersion of carbon dioxide leaking from underground sequestration, and outlier detection in eddy covariance data using extreme value theory.* ProQuest.

Shahid, N., Naqvi, I. H., & Qaisar, S. B. (2015). Characteristics and classification of outlier detection techniques for wireless sensor networks in harsh environments: a survey. *Artificial Intelligence Review*, *43*(2), 193–228.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437.

Storey, M. V., Van der Gaag, B., & Burns, B. P. (2011). Advances in on-line drinking water quality monitoring and early warning systems. *Water research*, *45*(2), 741–747.

Talagala, P., Hyndman, R., Smith-Miles, K., Kandanaarachchi, S., & Munoz, M. (2018). *Anomaly detection in streaming nonstationary temporal data* (Tech. Rep.). Monash University, Department of Econometrics and Business Statistics.

Talagala, P., & Hyndman, R. J. (2018). oddwater: A package for outlier detection in water quality sensor data [Computer software manual]. https://github.com/pridiltal/oddwater. DOI: 10.5281/zenodo.2890469.

Tang, J., Chen, Z., Fu, A. W.-C., & Cheung, D. W. (2002). Enhancing effectiveness of outlier detections for low density patterns. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 535–548).

Thottan, M., & Ji, C. (2003). Anomaly detection in ip networks. *IEEE Transactions on signal processing*, *51*(8), 2191–2204.

Tutmez, B., Hatipoglu, Z., & Kaymak, U. (2006). Modelling electrical conductivity of groundwater using an adaptive neuro-fuzzy inference system. *Computers & geosciences*, *32*(4), 421–433.

Weissman, I. (1978). Estimation of parameters and large quantiles based on the k largest observations. *Journal of the American Statistical Association*, *73*(364), 812–815.

Wilkinson, L. (2018). Visualizing big data outliers through distributed aggregation. *IEEE transactions on visualization and computer graphics*, *24*(1), 256–266.

Yu, J. (2012). A bayesian inference based two-stage support vector regression framework for soft sensor development in batch bioprocesses. *Computers & Chemical Engineering*, *41*, 134–144.

Zhang, K., Hutter, M., & Jin, H. (2009). A new local distance-based outlier detection approach for scattered real-world data. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 813–822).

Supporting Information for "A feature-based procedure for detecting technical outliers in water-quality data from in situ sensors"

Priyanga Dilini Talagala[1,2], Rob J. Hyndman[1,2], Catherine Leigh[1,3], Kerrie Mengersen[1,4], Kate Smith-Miles[1,5]

[1]ARC Centre of Excellence for Mathematics and Statistical Frontiers (ACEMS), Australia
[2]Department of Econometrics and Business Statistics, Monash University, Australia
[3]Institute for Future Environments, Science and Engineering Faculty, Queensland University of Technology, Australia
[4]School of Mathematical Sciences, Science and Engineering Faculty, Queensland University of Technology, Australia
[5]School of Mathematics and Statistics, University of Melbourne, Australia

Contents

Introduction

We considered the following outlier scoring techniques for the current work presented in this paper. The oddwater procedure can be easily updated with other unsupervised outlier scoring techniques.

Text S1.

NN-HD algorithm

This algorithm is inspired by the HDoutliers algorithm (Wilkinson, 2018) which is an unsupervised outlier detection algorithm that searches for outliers in high dimensional data assuming there is a large distance between outliers and the typical data. Nearest neighbor distances between points are used to detect outliers. However, variables with large variance can bring disproportional influence on Euclidean distance calculation. Therefore, the columns of the data sets are first normalized such that the data are bounded by the unit hyper-cube. The nearest neighbor distances are then calculated for each observation. In contrast to the implementation of HDoutliers algorithm available in the `HDoutliers` package (Fraley, 2018) our implementation available through the `oddwater` package now generates outlier scores instead of labels for each observation.

KNN-AGG and KNN-SUM algorithms

The NN-HD algorithm uses only nearest neighbor distances to detect outliers under the assumption that any outlying point present in the data set is isolated. For example, if there are two outlying points that are close to one another, but are far away from the rest of the valid data points, then the two outlying points become nearest neighbors to one another and give a small nearest neighbor distance for each outlying point. Because the NN-HD algorithm is dependent on the nearest neighbor distances, and the two outlying points do not show any significant deviation from other typical points with respect to nearest neighbor distance, the NN-HD algorithm now fails to detect these points as outliers.

Corresponding author: Priyanga Dilini Talagala, `dilini.talagala@monash.edu`

Following Angiulli and Pizzuti (2002), Madsen (2018) proposed two algorithms: aggregated k-nearest neighbor distance (KNN-AGG); and sum of distance of k-nearest neighbors (KNN-SUM) to overcome this limitation by incorporating k nearest neighbor distances for the outlier score calculation. The algorithms start by calculating the k nearest neighbor distances for each point. The k-dimensional tree (kd-tree) algorithm (Bentley, 1975) is used to identify the k nearest neighbors of each point in a fast and efficient manner. A weight is then calculated using the k nearest neighbor distances and the observations are ranked such that outliers are those points having the largest weights. For KNN-SUM, the weight is calculated by taking the summation of the distances to the k nearest neighbors. For KNN-AGG, the weight is calculated by taking a weighted sum of distances to k nearest neighbors, assigning nearest neighbors higher weight relative to the neighbors father apart.

LOF algorithm

The Local Outlier Factor (LOF) algorithm (Breunig et al., 2000) calculates an outlier score based on how isolated a point is with respect to its surrounding neighbors. Data points with a lower density than their surrounding points are identified as outliers. The local reachable density of a point is calculated by taking the inverse of the average readability distance based on the k (user defined) nearest neighbors. This density is then compared with the density of the corresponding nearest neighbors by taking the average of the ratio of the local reachability density of a given point and that of its nearest neighbors.

COF algorithm

One limitation of LOF is that it assumes that the outlying points are isolated and therefore fails to detect outlying clusters of points that share few outlying neighbors if k is not appropriately selected (Tang et al., 2002). This is known as a masking problem (Hadi, 1992), i.e. LOF assumes both low density and isolation to detect outliers. However, isolation can imply low density, but the reverse does not always hold. In general, low density outliers result from deviation from a high density region and an isolated outlier results from deviation from a connected dense pattern. Tang et al. (2002) addressed this problem by introducing a Connectivity-based Outlier Factor (COF) that compares the average chaining distances between points subject to outlier scoring and the average of that of its neighboring to their own k-distance neighbors.

INFLO algorithm

Detection of outliers is challenging when data sets contain adjacent multiple clusters with different density distributions (Jin et al., 2006). For example, if a point from a sparse cluster is close to a dense cluster, this could be misclassified as an outlier with respect to the local neighborhood as the density of the point could be derived from the dense cluster instead of the sparse cluster itself. This is another limitation of LOF (Breunig et al., 2000). The Influenced Outlierness (INFLO) algorithm (Jin et al., 2006) overcomes this problem by considering both the k nearest neighbors (KNNs) and reverse nearest neighbors (RNNs), which allows it to obtain a better estimation of the neighborhood's density distribution. The RNNs of an object, p for example, are essentially the objects that have p as one of their k nearest neighbors. Distinguish typical points from outlying points is helpful because they have no RNNs. To reduce the expensive cost incurred by searching a large number of KNNs and RNNs, the kd-tree algorithm was used during the search process.

LDOF algorithm

The Local Distance-based Outlier Factor (LDOF) algorithm (Zhang et al., 2009) also uses the relative location of a point to its nearest neighbors to determine the degree to which the point deviates from its neighborhood. LDOF computes the distance for an observation to its k-nearest neighbors and compares the distance with the average distances of the point's nearest neighbors. In contrast to LOF (Breunig et al., 2000), which uses local density, LDOF now uses relative distances to quantify the deviation of a point from its neighborhood system. One of the main differences between the two approaches (LDOF and LOF) is that LDOF represents the typical pattern of the data set by scattered points rather than crowded main clusters as in LOF (Zhang et al., 2009).

RKOF algorithm with Gaussian kernel

According to Gao, Hu, Zhang, Zhang, and Wu (2011), LOF is not accurate enough to detect outliers in complex and large data sets. Furthermore, the performance of LOF depends on the parameter k that determines the scale of the local neighborhood. The Robust Kernel-based Outlier Factor (RKOF) algorithm (Gao et al., 2011) tries to overcome these problems by incorporating variable kernel density estimates to address the first problem and weighted neighborhood density estimates to address the second problem. A Gaussian kernel with a bandwidth of $k-distance$ was used for density estimation. The two parameters: multiplication parameter for $k-distance$ of neighboring observations and sensivity parameter for $k-distance$ were set to 1 (default value given in Gao et al. (2011)).

References

Angiulli, F., & Pizzuti, C. (2002). Fast outlier detection in high dimensional spaces. In *European conference on principles of data mining and knowledge discovery* (pp. 15–27).

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, *18*(9), 509–517.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). Lof: identifying density-based local outliers. In *Acm sigmod record* (Vol. 29, pp. 93–104).

Fraley, C. (2018). Hdoutliers: Leland wilkinson's algorithm for detecting multidimensional outliers [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=HDoutliers` (R package version 1.0)

Gao, J., Hu, W., Zhang, Z. M., Zhang, X., & Wu, O. (2011). Rkof: robust kernel-based local outlier detection. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 270–283).

Hadi, A. S. (1992). Identifying multiple outliers in multivariate data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 761–771.

Jin, W., Tung, A. K., Han, J., & Wang, W. (2006). Ranking outliers using symmetric neighborhood relationship. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 577–593).

Madsen, J. H. (2018). Ddoutlier: Distance and density-based outlier detection [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=DDoutlier` (R package version 0.1.0)

Tang, J., Chen, Z., Fu, A. W.-C., & Cheung, D. W. (2002). Enhancing effectiveness of outlier detections for low density patterns. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 535–548).

Wilkinson, L. (2018). Visualizing big data outliers through distributed aggregation.

IEEE transactions on visualization and computer graphics, *24*(1), 256–266.

Zhang, K., Hutter, M., & Jin, H. (2009). A new local distance-based outlier detection approach for scattered real-world data. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 813–822).

Chapter 5

A Framework for Automated Anomaly Detection in High Frequency Water-Quality Data From *in situ* Sensors

This article is published in the *Science of the Total Environment*. The work is based on the collaborative research project carried out with the Queensland University of Technology and the Queensland Department of Environment and Science, Great Barrier Reef Catchment Loads Monitoring Program from April to July 2018.

Full paper is available at - https://www.sciencedirect.com/science/article/pii/S0048969719305662

Chapter 6

Conclusion

This thesis by publication is built around four articles. Although the four articles have their own focus motivated by a wide range of different analytical challenges from different fields, none of them is completely an anomaly. The four articles move around a unifying theme on anomaly detection in streaming time series data, with a different degree of attention to the common theme.

6.1 Summary of the Results and Contributions

Despite the long history of research on anomaly detection, the problem is still challenging owing to the evolving nature of the problem setting introduced by different applications and user requirements. This thesis is an attempt to reduce this gap by introducing three new algorithms, stray, oddstream and oddwater, for anomaly detection in temporal data with applications in pedestrian monitoring, security monitoring and sensor quality monitoring, respectively. The three algorithms stem from the analytical challenges introduced by the applications with various input data structures, definitions, problem specifications, user requirements, limitations of the state-of-the-art methods and unavailability of techniques that accommodate some of the data challenges.

6.1.1 The stray algorithm

Anomaly detection in high-dimensional data is a challenging yet important task, because it has applications in many fields. The HDoutliers algorithm by Wilkinson (2018) is a powerful algorithm for anomaly detection in high-dimensional data with a strong theoretical foundation. However, it suffers from a few limitations since it limits the anomalous score calculation only to the nearest neighbour distances and uses the Leader algorithm to form several clusters of points prior to anomalous score calculation. The effect of these limitations is a tendency to reduce computational efficiency and increase false detection rates under certain circumstances. Therefore, the main objective of Chapter 2 was to propose solutions to the limitation of the HDoutliers algorithm and thereby improve its capabilities.

The proposed algorithm, stray, addresses the limitations of the HDoutliers algorithm. In the stray algorithm, an anomaly is defined as an observation that deviates markedly from the majority with a large distance gap. It calculates an anomalous score for each data instance using k-nearest neighbour distances with the maximum gap. An approach based on extreme value theory is then applied to the anomalous scores to calculate a data-driven anomalous threshold. This improved algorithm can assign both a label and an anomalous score that explains the level of outlierness of each data instance.

This study offers two fundamental contributions. First, it proposes an improved algorithm for anomaly detection in high-dimensional data that addresses the limitations of the state-of-art-method, the HDoutliers algorithm. It outperforms the state-of-the-art method in both accuracy and computational efficiency. Among many other advantages, the stray algorithm has the ability to deal with the masking problem, multimodal distributions and inliers and outliers. The stray algorithm is specially designed for high-dimensional data. As the second contribution, the study demonstrates how the stray algorithm can assist in detecting anomalies present in other data structures, such as temporal data and streaming data, using feature engineering.

Since the stray algorithm is based on the distance definition of an anomaly, the algorithm expects data instances to have a clear distance separation between the anomalous and

typical points. Then, only the anomalous points (if any) have significantly large k-nearest neighbour distances with the maximum gap that discriminate anomalies from typical points. However, some applications do not exhibit large gaps between typical points and anomalies. Instead, the anomalies deviate from the majority, or the region of typical data, gradually, without introducing a large distance between typical and anomalous points. In the absence of clear distance separation between anomalous points and the typical points, the stray algorithm fails to detect anomalies since distance measures are the primary source of information for the algorithm to detect anomalies. This limitation of the stray algorithm motivates the second algorithm proposed in Chapter 3 of this thesis.

6.1.2 The oddstream algorithm

In addition to the aforementioned limitation of the stray algorithm, the limited research attempts for detecting anomalous series within a large collection of streaming data motivated the second algorithm of this thesis, the oddstream algorithm. The primary focus of Chapter 3 was to develop a powerful automated method to detect anomalous series within a large collection of series in the streaming data context.

In the oddstream algorithm, an anomaly is defined as an observation that is very unlikely, given the recent distribution of a given system. In this algorithm, a boundary for the system's typical behaviour is defined using extreme value theory. Then, a sliding window is used to test newly arrived data. The model uses time series features as inputs and a density-based comparison to locate nonstationarity. This algorithm can detect significant changes in the typical behaviour and automatically update the anomalous threshold on detecting nonstationarity.

This study offers three fundamental contributions. First, it proposes a new framework that provides early detection of anomalies within a large collection of streaming time series data. Second, it proposes a novel approach that adapts to nonstationarity. Third, using various synthetic and real datasets, it demonstrates the wide applicability and usefulness of the algorithm. Application of the oddstream algorithm with data obtained using fibre optic cables for intrusion detection showed that the algorithm has the ability to deal with large nonstationary streaming data that may have multimodal distributions.

6.1.3 The oddwater algorithm

Automated *in situ* sensors have the potential to revolutionise the way we monitor environmental conditions. However, the data produced by these sensors are prone to errors because of many reasons, such as miscalibration, biofouling and battery failures (Horsburgh et al., 2015). These technical outliers make the data unreliable for scientific analysis. Therefore, to ensure water-quality sensors yield high-quality data, we need to automate the real-time detection of technical outliers in such data. However, a customised method to detect technical outliers in water-quality data from *in situ* sensors is lacking. No exiting outlier detection method is able to address this challenge owing to the complex nature of the definition of a technical outlier in water-quality data from *in situ* sensors. Therefore, the main objective of Chapters 4 and 5 was to propose a new framework to detect technical outliers in high-frequency water-quality data from *in situ* sensors.

This study proposes an automated framework that provides early detection of technical outliers, caused by technical issues, in water-quality data from *in situ* sensors. We compare two approaches to this problem: (1) using forecasting models (Chapter 5) and (2) using feature vectors with extreme value theory (Chapter 4). In the forecasting models, observations are identified as outliers when they fall outside the bounds of an established prediction interval. Two strategies are considered for this comparison study: anomaly detection (AD) and anomaly detection and mitigation (ADAM) for the detection process. With ADAM, the detected outliers are replaced with the forecast prior to the next prediction, whereas AD simply uses the previous measurements without altering detected outliers. The feature-based framework first identifies the data features that differentiate outlying instances from typical behaviours. Then, statistical transformations are applied to make the outlying instances stand out in transformed data space. Unsupervised outlier scoring techniques are then applied to the transformed data space. An approach based on extreme value theory is used to calculate a threshold for each potential outlier. The proposed frameworks are evaluated using two datasets obtained from *in situ* sensors in rivers flowing into the Great Barrier Reef lagoon.

The feature-based approach (in Chapter 4) successfully identified outliers involving abrupt changes in turbidity, conductivity and river level, including sudden spikes, sudden isolated drops and level shifts, while maintaining very low false detection rates. Since this is an unsupervised algorithm, it can be easily extended to other water-quality variables, other sites and also to other outlier detection tasks in other application domains. The only requirement is to select suitable transformation methods according to the data features that differentiate the outlying instances from the typical behaviours of a given system. The transformations used in this study were mainly chosen as appropriate to the data collected from Sandy Creek and Pioneer River. Domain-specific knowledge plays a vital role when selecting suitable transformations.

6.2 Future Work

Since this is a thesis by publication, each article should be self-contained and therefore has been published with all the relevant possible further research directions discussed in detail. To avoid repetition, this section summarises only the key future research priorities, which are deemed underrepresented in the current literature.

While the HDoutliers algorithm is powerful, several classes of counterexamples were identified where the structural properties of the data did not enable the HDoutliers algorithm to detect certain types of outliers. However, I acknowledge that these counterexamples are not diverse and challenging enough to generalise the findings to conclude that stray is always the superior algorithm. Therefore, an important open research problem is to assess the effectiveness of these algorithms across the broadest possible problem space defined by different datasets with diverse properties (Kang, Hyndman, and Smith-Miles, 2017). It would also be interesting to explore how other classes of problems with various structural properties can influence the performance of the stray algorithm and where its weaknesses might lie. This type of instance space analysis (Smith-Miles et al., 2014) will enable further insights into improved algorithm design.

In the oddstream algorithm, the use of the feature-based representation of time series is recommended, owing to its many advantages over the instance-based representation of time series. In the present study, only 14 features were used to represent a given time

series. Further exploration of feature extraction and automatic feature selection methods is required to create a richer feature space that is suitable for many applications in the streaming data context. The proposed algorithm uses the first two principal components to obtain a two-dimensional feature space, and then defines an anomalous threshold on the resulting two-dimensional feature space. It is expected that in further studies, other dimension reduction techniques will be used, such as multidimensional scaling and random projection, to investigate the effects of such techniques on the performance of the proposed framework. Further, the density estimation in the proposed algorithm was performed using a bivariate kernel density estimation method. Since the density values in the tail are used to build the model of the typical behaviour, additional experiments need to be conducted on density estimation methods, to improve the tail estimation. On this topic, the log-spline bivariate density estimation method and the local likelihood density estimation method will be considered, with the aim of achieving a better tail estimation, and thereby improving the performance of the proposed algorithm. The current algorithm is developed under the assumption that the measurements produced by sensors are one-dimensional. Rapid advances in hardware technology have made it possible for many sensors to capture multiple measurements simultaneously, leading ultimately to a collection of multidimensional multivariate streaming time series data. Therefore, an important open research problem is to extend the oddstream algorithm to handle multidimensional, multivariate streaming data. Extending the oddstream algorithm to the detection of anomalies in this data context may allow us to perform anomaly detection in an even wider range of application domains.

Spatiotemporal anomaly detection for water-quality data also lags behind that for other *in situ* sensor data types (e.g., air quality or meteorology; Wu, Liu, and Chawla, 2008) because river data pose new challenges, such as the complex relationships between neighbouring sensors due to branching networks and flow directionality, tendency for biofouling and the highly dynamic nature of river water even under typical conditions (Kang et al., 2009). These challenges make traditional anomaly detection approaches inadequate for spatiotemporal water-quality data and require new methods. The oddwater algorithm is expected to expand into space and time so that it can deal with the spatiotemporal

correlation structure along branching river networks. This will in turn provide a fundamental step-change in scientific understanding of the spatiotemporal dynamics of water quality in rivers and their networks and the potential downstream effects of pollutant loads.

6.3 Research Reproducibility

Research reproducibility is an important topic in modern science because it provides a general schema and an infrastructure to regenerate quantitative scientific results using the original datasets and methods (Stodden, Leisch, and Peng, 2014). Therefore, to facilitate reproducibility and reuse of the results presented in this thesis, I undertook several actions under the three key areas: software, data and papers (Stodden, Leisch, and Peng, 2014).

6.3.1 Software

This thesis introduces three R packages for anomaly detection.

The first R package is an accompaniment to the algorithm proposed in Chapter 2 and includes useful functions for detecting anomalies in high-dimensional data. Version 1.0.0.9000 of the package was used for the results presented in Chapter 2 and is available from GitHub at `https://github.com/pridiltal/stray`.

The second R package, `oddstream`, is an accompaniment to the algorithm proposed in Chapter 3 and includes useful functions for detecting anomalous series within a large collection of streaming time series data. Version 0.5.0 of the package was used for the results presented in Chapter 3 and is available from GitHub at `https://github.com/pridiltal/oddstream`.

The third package is an accompaniment to the algorithm proposed in Chapters 4 and 5 and includes useful functions for detecting technical anomalies in water-quality data from *in situ* sensors. Version 0.5.0.9000 of the package was used for the results presented in Chapters 4 and 5 and is available from GitHub at `https://github.com/pridiltal/oddwater`.

6.3.2 Data

All the datasets on which the results are computed in each article are available via the corresponding R package. A Shiny web application available through the `oddwater` R package provides greater visual insight into the water-quality data from *in situ* sensors and was heavily used during the labelling process to pinpoint observations.

6.3.3 Papers

The three main articles in Chapters 2, 3 and 4 describe the corresponding algorithms in detail and compare their implementations using various datasets. The source files, including datasets and R code to reproduce all figures, tables and analysis of each article, can be found in the following public GitHub repositories.

Chapter 2: 'Anomaly Detection for High Dimensional Data' at https://github.com/pridiltal/stray_manuscript.

Chapter 3: 'Anomaly Detection in Streaming Non-stationary Temporal Data' at https://github.com/pridiltal/oddstream_manuscript.

Chapters 4: 'A Feature-Based Procedure for Detecting Technical Outliers in Water-Quality Data from *in situ* Sensors' at https://github.com/pridiltal/oddwater_manuscript.

These articles were written entirely using `Rmarkdown` (Allaire et al., 2019), and compiled into a thesis using the `bookdown` R package (Xie, 2019), with the Monash PhD thesis rmarkdown template available at https://github.com/robjhyndman/MonashThesis. The source files of this thesis are available at https://github.com/pridiltal/PhD_Thesis_2019.

Bibliography

Abuzaid, A, A Hussin, and I Mohamed (2013). Detection of outliers in simple circular regression models using the mean circular error statistic. *Journal of Statistical Computation and Simulation* **83**(2), 269–277.

Allaire, J, Y Xie, J McPherson, J Luraschi, K Ushey, A Atkins, H Wickham, J Cheng, W Chang, and R Iannone (2019). *rmarkdown: Dynamic Documents for R*. R package version 1.13. https://rmarkdown.rstudio.com.

Ben-Gal, I (2005). "Outlier detection". In: *Data Mining and Knowledge Discovery Handbook*. Springer, pp.131–146.

Burridge, P and AMR Taylor (2006). Additive outlier detection via extreme-value theory. *Journal of Time Series Analysis* **27**(5), 685–701.

Chandola, V, A Banerjee, and V Kumar (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)* **41**(3), 15.

Clifton, DA (2009). "Novelty detection with extreme value theory in jet engine vibration data". PhD thesis. University of Oxford.

Clifton, DA, S Hugueny, and L Tarassenko (2011). Novelty detection with multivariate extreme value statistics. *Journal of Signal Processing Systems* **65**(3), 371–389.

Coles, S (2001). *An Introduction to Statistical Modeling of Extreme Values*. Lecture Notes in Control and Information Sciences. Springer.

Embrechts, P, C Klüppelberg, and T Mikosch (2013). *Modelling Extremal Events: for Insurance and Finance*. Stochastic Modelling and Applied Probability. Springer Berlin Heidelberg.

Faria, ER, IJ Gonçalves, AC de Carvalho, and J Gama (2016). Novelty detection in data streams. *Artificial Intelligence Review* **45**(2), 235–269.

Fisher, RA and LHC Tippett (1928). Limiting forms of the frequency distribution of the largest or smallest member of a sample. In: *Mathematical Proceedings of the Cambridge Philosophical Society*. Vol. 24. 02. Cambridge Univ Press, pp.180–190.

Fulcher, BD and NS Jones (2014). Highly comparative feature-based time-series classification. *IEEE Transactions on Knowledge and Data Engineering* **26**(12), 3026–3037.

Fulcher, BD, MA Little, and NS Jones (2013). Highly comparative time-series analysis: the empirical structure of time series and their methods. *Journal of the Royal Society Interface* **10**(83), 20130048.

Fulcher, BD (2012). "Highly comparative time-series analysis". PhD thesis. University of Oxford.

Galambos, J, J Lechner, and E Simiu (2013). *Extreme Value Theory and Applications: Proceedings of the Conference on Extreme Value Theory and Applications, Volume 1 Gaithersburg Maryland 1993*. Extreme Value Theory and Applications: Proceedings of the Conference on Extreme Value Theory and Applications, Gaithersburg, Maryland, 1993. Springer US.

Gama, J (2010). *Knowledge Discovery from Data Streams*. Chapman and Hall/CRC.

Grubbs, FE (1969). Procedures for detecting outlying observations in samples. *Technometrics* **11**(1), 1–21.

Gupta, M, J Gao, CC Aggarwal, and J Han (2014). Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and Data Engineering* **26**(9), 2250–2267.

Hofmann, H, H Wickham, and K Kafadar (2017). Value plots: Boxplots for large data. *Journal of Computational and Graphical Statistics* **26**(3), 469–477.

Horsburgh, JS, SL Reeder, AS Jones, and J Meline (2015). Open source software for visualization and quality control of continuous hydrologic and water quality sensor data. *Environmental Modelling & Software* **70**, 32–44.

Hugueny, S (2013). "Novelty detection with extreme value theory in vital-sign monitoring". PhD thesis. University of Oxford.

Hyndman, RJ (1996). Computing and graphing highest density regions. *The American Statistician* **50**(2), 120–126.

Hyndman, RJ, E Wang, and N Laptev (2015). Large-scale unusual time series detection. In: *2015 IEEE international conference on data mining workshop (ICDMW)*. IEEE, pp.1616–1619.

Kang, JM, S Shekhar, M Henjum, PJ Novak, and WA Arnold (2009). Discovering teleconnected flow anomalies: A relationship analysis of dynamic neighborhoods (RAD) approach. In: *International Symposium on Spatial and Temporal Databases*. Springer, pp.44–61.

Kang, Y, RJ Hyndman, and K Smith-Miles (2017). Visualising forecasting algorithm performance using time series instance spaces. *International Journal of Forecasting* **33**(2), 345–358.

Kumar, D, JC Bezdek, S Rajasegarar, M Palaniswami, C Leckie, J Chan, and J Gubbi (2016). Adaptive cluster tendency visualization and anomaly detection for streaming data. *ACM Transactions on Knowledge Discovery from Data (TKDD)* **11**(2), 24.

Lavin, A and S Ahmad (2015). Evaluating real-time anomaly detection algorithms–the numenta anomaly benchmark. In: *Machine Learning and Applications (ICMLA), 2015 IEEE 14th International Conference on*. IEEE, pp.38–44.

Pinto, C and P Garvey (2016). *Advanced Risk Analysis in Engineering Enterprise Systems*. Statistics: A Series of Textbooks and Monographs. CRC Press.

Schwarz, KT (2008). "Wind dispersion of carbon dioxide leaking from underground sequestration, and outlier detection in eddy covariance data using extreme value theory". PhD thesis.

Smith-Miles, K, D Baatar, B Wreford, and R Lewis (2014). Towards objective measures of algorithm performance across instance space. *Computers & Operations Research* **45**, 12–24.

Stodden, V, F Leisch, and RD Peng (2014). *Implementing Reproducible Research*. CRC Press.

Wang, X, K Smith, and RJ Hyndman (2006). Characteristic-based clustering for time series data. *Data Mining and Knowledge Discovery* **13**(3), 335–364.

Weissman, I (1978). Estimation of parameters and large quantiles based on the k largest observations. *Journal of the American Statistical Association* **73**(364), 812–815.

Wilkinson, L (2018). Visualizing big data outliers through distributed aggregation. *IEEE Transactions on Visualization and Computer Graphics* **24**(1), 256–266.

Wu, E, W Liu, and S Chawla (2008). Spatio-temporal outlier detection in precipitation data. In: *International Workshop on Knowledge Discovery from Sensor Data*. Springer, pp.115–133.

Xie, Y (2019). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.11. https://github.com/rstudio/bookdown.